

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

ROBÔS MODULARES BASEADOS EM AGENTES MECATRÔNICOS

por

Anselmo Rafael Cukla

Tese para obtenção do Título de
Doutor em Engenharia

Porto Alegre, Setembro de 2016

ROBÔS MODULARES BASEADOS EM AGENTES MECATRÔNICOS

por

Anselmo Rafael Cukla

Mestre em Engenharia

Tese submetida ao Programa de Pós-Graduação em Engenharia Mecânica, da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos necessários para a obtenção do Título de

Doutor em Engenharia

Área de Concentração: Processos de Fabricação

Orientador: Prof. Dr. Eduardo André Perondi

Co-orientador: Prof. Dr. José Antônio Barata de Oliveira

Comissão de Avaliação:

Profª. Dra. Silvia Silva da Costa BotelhoPPGQVS / FURG

Prof. Dr. Flávio José Lorini PROMEC / UFRGS

Prof. Dr. Gilberto Dias da Cunha DEMEC / UFRGS

Prof. Dr. Jakson Manfredini Vassoler

Coordenador do PROMEC

Porto Alegre, 22 de Setembro de 2016

Dedico este trabalho à pessoa mais importante da minha vida, Marisa.

AGRADECIMENTOS

Agradeço a Deus, pela sabedoria e à força e de continuar nos momentos difíceis.

À CAPES e ao PROMEC, pela bolsa de estudos concedida e por proporcionar um ambiente favorável ao desenvolvimento intelectual.

A minha esposa Marisa, que com muita paciência e carinho, me acompanho durante o mestrado e o doutorado.

Aos meus pais, Rodolfo e Luisa, que confiaram em mim e colaboraram com amor e apoio incondicional e assim poder concluir este trabalho.

Aos meus orientadores da tese Prof. Dr. Eduardo André Perondi e Prof. Dr. José Barata, pela confiança depositada, amizade, dedicação e apoio em as etapas do desenvolvimento desta tese, sendo ainda, grandes exemplos de profissionais da educação o qual desejo seguir.

Aos colegas do GPFAI, em especial a Rafael C. Izquierdo, pela ajuda incondicional em diversas etapas do trabalho.

Aos colegas do LAMECC, em especial ao Alexandre N. Stedile pelo suporte nas diversas etapas do trabalho.

RESUMO

Nas linhas de montagens industriais, a fim de atender os requisitos de mercado e de ciclo de vida dos produtos, os requisitos de manufatura e as novas tecnologias presentes nos equipamentos indicam a necessidade de reconfiguração e reprogramação do fluxo de processos de forma cada vez mais frequente. Atualmente, uma das opções para implantar um sistema de manufatura flexível, capaz de reagir às mudanças que ocorrem no processo de fabricação, consiste na utilização de tecnologias que forneçam maior flexibilidade, capacidade de reutilização e menor custo. Neste contexto, os robôs baseados em módulos mecatrônicos podem ser uma alternativa em relação aos manipuladores convencionais, pois apresentam uma estrutura cinemática flexível, podendo se adaptar às mudanças das linhas de produção, nas indústrias de manufatura. O presente trabalho apresenta uma proposta para o desenvolvimento de módulos mecatrônicos para a montagem de robôs manipuladores modulares, baseada em um procedimento sequencial composto das seguintes etapas: (a) elaboração do projeto mecânico modular; (b) projeto dos sistemas eletrônicos e de atuação para cada módulo; (c) definição dos agentes mecatrônicos; e (d) descrição dos modelos matemáticos e os algoritmos de comunicação entre módulos mecatrônicos. Nesta pesquisa apresenta-se um estudo no qual os módulos mecatrônicos utilizam energia de origem pneumática e são constituídos por unidades independentes utilizadas na formação de estruturas robotizadas as quais permitem a montagem de diferentes arquiteturas. Um estudo de caso é apresentado para ilustrar a construção de um robô modular cartesiano. Este robô é construído por meio de acoplamentos de módulos mecatrônicos e gerenciado pela associação dos agentes mecatrônicos presentes no sistema, os quais equacionam a cinemática da estrutura formada, planejam a trajetória a ser executada e disponibilizam informações que podem ser utilizadas para o controle, supervisão e proteção do sistema por exemplo. A arquitetura proposta permite a reconfiguração dos recursos de hardware e software, de forma que todos os módulos do robô podem ser reorganizados e/ou substituídos, dependendo da função, aplicação para as quais se destinam.

Palavras-chave: Módulos Mecatrônicos; Agentes; Robôs Manipuladores; Robôs Modulares.

ABSTRACT

In industrial manufacturing lines, in order to meet the market requirements and life cycle of manufactured products, the manufacturing requirements and the present of new technologies in equipment, indicate the need for reconfiguration and reprogramming processes, which are becoming more frequent. Currently, one of the options to deploy a flexible manufacturing system that is capable of reacting to changes in the manufacturing process is the use of technologies that provide greater flexibility, reusability and lower cost. In this context, the robots based on mechatronic modules can be an alternative to conventional manipulators, since they have a flexible kinematic structure, which can adapt to the changes in production lines in manufacturing industries. This paper presents a proposal for the development of mechatronic modules for assembly robots modular manipulators, based on a sequential procedure consists of the following steps: (a) Develop a modular mechanical design; (b) design electronic systems and operations for each module; (c) definition of mechatronic agents; and (d) a description of mathematical models and algorithms of the communication between mechatronic modules. This research presents a study where the mechatronic modules use pneumatic energy and consist of independent units used in the formation of robotic structures, thus allowing the assembly of different architectures. In a case study, the construction of a modular Cartesian robot is presented. This robot is built by mounting the mechatronic modules and is managed by mechatronic agents present in the system (Multi-Agent System). This system obtains the kinematic equations of the formed structure, realize the path planning, and provide information that can be used for the control, like supervision and protection system for example. The proposed architecture allows reconfiguration of hardware and software resources, so that all robot modules can be rearranged and/or replaced, depending on the function or, the final application.

Keywords: Mechatronic Modules, Agents, Manipulator Robots, Modular Robots.

ÍNDICE

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO..... | 1 |
| 1.1 | Motivação | 2 |
| 1.2 | Descrição do problema | 2 |
| 1.3 | Metodologia | 3 |
| 1.4 | Objetivo do estudo | 4 |
| 1.5 | Principais contribuições | 5 |
| 1.6 | Organização do trabalho | 5 |
| 2 | REVISÃO BIBLIOGRÁFICA | 6 |
| 2.1 | Fundamentação teórica | 6 |
| 2.1.1 | Estruturas cinemáticas de robôs clássicos | 6 |
| 2.1.2 | Conceito e tipo de agentes | 7 |
| 2.1.3 | Modulo Mecatrônico e Agente Mecatrônico | 13 |
| 2.1.4 | Componentes pneumáticos e eletropneumáticos | 17 |
| 2.1.5 | Sistema robótico modular | 19 |
| 2.2 | Estado da Arte..... | 22 |
| 2.2.1 | Robôs modulares e agentes mecatrônicos..... | 24 |
| 3 | MÓDULOS MECATRÔNICOS BASEADOS EM AGENTES | 35 |
| 3.1 | Metodologia de projeto dos módulos mecatrônicos | 36 |
| 3.1.1 | Especificações do projeto | 36 |
| 3.1.2 | Projeto Conceitual | 37 |
| 3.1.3 | Projeto Preliminar | 41 |
| 3.1.4 | Acoplamento entre módulos mecatrônicos..... | 47 |
| 3.2 | Definição da cinemática da estrutura formada pelos módulos mecatrônicos..... | 49 |
| 3.2.1 | Cinemática Direta | 50 |
| 3.2.2 | Determinação da matriz de transformação | 51 |
| 3.3 | Proposta para o planejamento de trajetórias controle para robôs modulares..... | 52 |
| 3.3.1 | Planejamento de trajetórias para robôs manipuladores | 53 |
| 3.3.2 | Controle dos módulos mecatrônicos..... | 56 |
| 3.4 | Os agentes e os módulos mecatrônicos | 59 |
| 3.4.1 | Caracterização dos Módulos Mecatrônicos | 59 |
| 3.4.2 | Definição da Estrutura Robótica..... | 66 |
| 3.4.3 | Agentes da plataforma | 74 |
| 3.5 | Considerações finais | 77 |
| 4 | ROBÔS MODULARES | 80 |
| 4.1 | Montagem do robô manipulador | 80 |
| 4.1.1 | Elementos para a montagem de um robô modular cartesiano | 81 |
| 4.1.2 | Descrição dos tipos de conexões entre as unidades | 82 |
| 4.2 | Características da cinemática de um robô cartesiano | 85 |
| 4.2.1 | Cinemática direta | 85 |
| 4.2.2 | Cinemática inversa de um robô cartesiano usando método iterativos | 88 |
| 4.3 | Dinâmica de um robô rígido | 90 |
| 4.3.1 | Modelo dinâmico de um robô com elos rígidos | 91 |
| 4.3.2 | Controle Independente por Junta | 91 |
| 4.4 | Planejamento de trajetórias aplicadas a robôs modulares..... | 96 |

| | | |
|----------|---|------------|
| 4.5 | Proposta do sistema Multiagentes em módulos mecatrônicos | 97 |
| 4.5.1 | Java Agent DEvelopment – JADE | 98 |
| 4.6 | Considerações finais | 103 |
| 5 | ESTUDO DE CASO E PROCEDIMENTO SISTEMÁTICO PARA DEFINIÇÃO DE ROBÔS MODULARES..... | 105 |
| 5.1 | Estudo de caso: robô modular cartesiano | 105 |
| 5.2 | Estudo de caso de um robô cartesiano de acionamento pneumático | 108 |
| 5.2.1 | Simulação de Planejamento de trajetória..... | 108 |
| 5.2.2 | Controle aplicado ao seguimento de uma trajetória..... | 112 |
| 5.2.3 | Gerenciamento de robôs segundo o conceito de Sistemas Multiagentes..... | 115 |
| 5.3 | Considerações finais | 118 |
| 6 | CONCLUSÕES FINAIS E TRABALHOS FUTUROS | 123 |
| 6.1 | Conclusões..... | 123 |
| 6.2 | Trabalhos futuros | 125 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 126 |
| | ANEXO A – ALGORITMOS DE CONTROLE | 131 |
| A.1 | Controle PID | 131 |
| A.2 | Controle por realimentação de Estados | 132 |
| A.3 | Controladores por modos deslizantes | 134 |
| | ANEXO B – MÉTODOS ITERATIVOS DE NEWTON-RAPHSON..... | 136 |
| B.1 | Convergência do Método de Newton-Raphson | 137 |
| B.2 | Interpretação geométrica | 138 |

LISTA DE FIGURAS

| | | |
|-------------|--|----|
| Figura 2.1 | Modelo de um agente simples [Hama, 2012]. | 8 |
| Figura 2.2 | FIPA <i>Contract Net Interaction Protocol</i> . | 11 |
| Figura 2.3 | FIPA <i>Request Interaction Protocol</i> [FIPA, 2002]. | 12 |
| Figura 2.4 | Interação entre sistemas dentro de um módulo mecatrônico | 13 |
| Figura 2.5 | Sistema pneumático de posicionamento [Sarmanho, 2014]. | 18 |
| Figura 2.6 | Módulo ativo básico. | 21 |
| Figura 2.7 | Configuração modular de um robô SCARA. | 22 |
| Figura 2.8 | Dois tipos de configurações de robôs modulares. | 25 |
| Figura 2.9 | Arquitetura descentralizada proposta por Atta-Konadu et al., 2005. | 26 |
| Figura 2.10 | Diagrama de sincronização das unidades de controle. | 27 |
| Figura 2.11 | Conceito de formação de robôs modulares [Larizza et al., 2006]. | 28 |
| Figura 2.12 | Configuração modular cinemática paralelo-serial. | 29 |
| Figura 2.13 | Organização do software em função do hardware do robô modular. | 29 |
| Figura 2.14 | Arquitetura de software de um sistema Multiagentes. | 32 |
| Figura 3.1 | Módulos auxiliares utilizados em um robô modular cartesiano | 40 |
| Figura 3.2 | Estrutura serial de um robô cartesiano modular | 42 |
| Figura 3.3 | Módulo mecatrônico de acionamento linear. | 43 |
| Figura 3.4 | Módulo mecatrônico e os componentes internos | 44 |
| Figura 3.5 | Alocação dos componentes de um módulo mecatrônico | 44 |
| Figura 3.6 | Cilindro sem haste de atuação pneumático Festo | 45 |
| Figura 3.7 | ligação entre dois módulos mecatrônicos | 48 |
| Figura 3.8 | Módulo auxiliar tipo 1 | 48 |
| Figura 3.9 | Módulo auxiliar tipo 2 | 49 |
| Figura 3.10 | Módulo auxiliar tipo 3 | 49 |
| Figura 3.11 | Algoritmo de obtenção dos parâmetros de Denavit-Hartenberg. | 52 |
| Figura 3.12 | Etapas para a abordagem do problema proposto | 54 |
| Figura 3.13 | Interpolação <i>spline</i> de 3º ordem. | 55 |
| Figura 3.14 | Controle para cada módulo mecatrônico | 57 |
| Figura 3.15 | Diagrama de Blocos PID. | 58 |
| Figura 3.16 | Diagrama de blocos controle por modos deslizantes | 59 |
| Figura 3.17 | Arquitetura proposta: visão geral [adaptado de Cavalcante, 2012. | 60 |

| | | |
|-------------|---|-----|
| Figura 3.18 | Arquitetura proposta: cada agente cognitivo pode comunicar-se com vários agentes motores e outros agentes cognitivos. | 61 |
| Figura 3.19 | Agentes que compõem um módulo mecatrônico..... | 63 |
| Figura 3.20 | Exemplo de funcionalidades em agentes cognitivo e motor..... | 65 |
| Figura 3.21 | Agente cognitivo implementando um bloco funcional..... | 66 |
| Figura 3.22 | Hierarquia de agentes mecatrônicos chamando funcionalidades remotas. | 67 |
| Figura 3.23 | Coalizão de agentes mecatrônicos em um manipulador de 3GDL | 68 |
| Figura 3.24 | Mapeamento de módulo mecatrônico como agente simples (servoposicionador linear) | 69 |
| Figura 3.25 | Cada módulo mecatrônico do sistema pode ser mapeado diretamente a um agente motor..... | 70 |
| Figura 3.26 | Restrição de área para a interação dos agentes e formação de pagina amarela..... | 72 |
| Figura 3.27 | Diagrama orientado a objetos dos agentes..... | 74 |
| Figura 3.28 | Os agentes da arquitetura e seus relacionamentos. | 77 |
| Figura 4.1 | Modelo de robô cartesiano com acionamento pneumático proposto..... | 80 |
| Figura 4.2 | Robô modular e seus periféricos..... | 82 |
| Figura 4.3 | Conexão base com o 1º GDL de acionamento linear..... | 83 |
| Figura 4.4 | Conexões elétricas e dados | 84 |
| Figura 4.5 | Ligação explodida entre dois módulos mecatrônicos | 85 |
| Figura 4.6 | Ligação física entre um efetuator e uma unidade mecatrônica..... | 85 |
| Figura 4.7 | Etapas para a definição da matriz DH..... | 86 |
| Figura 4.8 | Algoritmo proposto para a execução da cinemática inversa de um robô modular serial..... | 90 |
| Figura 4.9 | Diagrama de blocos genérico de um sistema de controle independente por junta de um robô manipulador [adaptado de Oliveira e Lages, 2006]..... | 92 |
| Figura 4.10 | Ciclo de vida de um agente FIPA [FIPA, 2002]..... | 99 |
| Figura 4.11 | Execução de um agente JADE..... | 101 |
| Figura 5.1 | Sequência de montagem de um robô modular..... | 106 |
| Figura 5.2 | Componentes de uma plataforma de um robô modular | 107 |
| Figura 5.3 | Processo de geração de trajetória para robôs modulares | 109 |
| Figura 5.4 | Pontos intermediários e respectiva trajetória de 3º grau..... | 110 |
| Figura 5.5 | Curvas de posição da junta 1 do robô modular..... | 111 |
| Figura 5.6 | Curvas de posição da junta 2 do robô modular..... | 111 |

| | | |
|-------------|---|-----|
| Figura 5.7 | Curvas de posição da junta 3 do robô modular..... | 112 |
| Figura 5.8 | Controle Torque Calculado 1º GDL | 113 |
| Figura 5.9 | Controle Torque Calculado 2º GDL | 114 |
| Figura 5.10 | Controle Torque Calculado 3º GDL | 114 |
| Figura 5.11 | Inicialização do sistema computacional de gerenciamento | 115 |
| Figura 5.12 | Localização dos agentes no sistema..... | 116 |
| Figura 5.13 | Indicação dos módulos mecatrônicos que estão conectados à plataforma | 117 |
| Figura 5.14 | Módulos ativos conectados no sistema..... | 117 |
| Figura 5.15 | Abrangência e localização dos agentes da plataforma..... | 118 |
| Figura 5.16 | Comportamentos dos módulos mecatrônicos | 119 |
| Figura 5.17 | Comportamento cíclico de um agente (execução de um algoritmo de controle).. | 120 |
| Figura 5.18 | Negociação entre um módulo mecatrônico e um computador..... | 120 |
| Figura 5.19 | Comportamento reativo de um módulo mecatrônico..... | 121 |
| Figura 5.20 | Fluxograma de utilização da metodologia para a configuração de robôs modulares. | 122 |

LISTA DE TABELAS

| | | |
|------------|---|-----|
| Tabela 2.1 | Conceitos de um Robô Industrial abordado por diferentes organizações..... | 7 |
| Tabela 3.1 | Especificações técnicas dos atuadores DGPIIL [Festo, 2016]..... | 45 |
| Tabela 3.2 | Características do Raspberry Pi Modelo B | 46 |
| Tabela 3.3 | Características do microcontrolador PIC18F4580..... | 46 |
| Tabela 3.4 | Parâmetros de Denavit-Hartenberg de um sistema modular..... | 51 |
| Tabela 4.1 | Parâmetros cinemáticos | 86 |
| Tabela 4.2 | Matriz de DH para um robô cartesiano de 3º GDL..... | 87 |
| Tabela 4.3 | Valores limites das juntas do robô cartesiano..... | 88 |
| Tabela 5.1 | Coordenadas dos pontos do efetuador | 110 |
| Tabela 5.2 | Ganhos do controlador TC [adaptado de Sarmanho, 2014]..... | 113 |

LISTA DE SIGLAS E ABREVIATURAS

| | |
|---------|---|
| IFTToMM | International Federation for the Promotion of Mechanism and Machine Science |
| BARA | British Automation & Robot Association |
| RIA | Robotic Industries Association |
| ISO | International Organization for Standardization |
| AFNOR | Association Française de NORmalisation |
| PPP | 3 juntas prismáticas seriais |
| RPP | 1 junta rotacional e duas prismáticas |
| RRP | 2 juntas rotacionais e uma prismática |
| FIPA | Foundation for Intelligent Physical Agents |
| MAS | Multi-Agent System |
| ACL | Agent Communication Language specifications |
| CFP | Call for Proposal – mensagem de solicitação |
| AID | Identificador do Agente |
| CAN | Controller Area Network |
| SPI | Serial Peripheral Interface |
| AGV | Automated guided vehicle |
| OROCOS | Open RObot COntrol Software |
| AMS | sistema de gerenciamento de agentes |
| DF | Diretorio facilitador |
| MTS | sistema de transporte de mensagem |
| GDL | Grau de liberdade |
| N | Newton |
| Vcc | Tensão em corrente continua |
| Pa | Pascal |
| D-H | Denavit Hartenberg |
| SCARA | Selective Compliant Articulated Robot Arm |
| SMC | Controlador Slide Mode Control |
| PID | Controlador Proporcional Integral Derivativo |
| Kp | Ganho proporcional |

| | |
|-------|-----------------------------------|
| K_d | Ganho derivativo |
| K_i | Ganho integral |
| RA | Agente Recurso |
| CLA | Agente Líder de Coalisão |
| ASk | Habilidades Atômicas |
| CSk | Habilidades Compostas |
| YPA | Agente paginas amarelas |
| DA | Agente de distribuição |
| T | Matriz de transformação homogênea |
| J | Matriz Jacobiana |
| H | Matriz de Inercias |
| C | Matriz de coriolis |
| G | Matriz de gravidade |
| e | Erro do controle |
| TC | Algoritmo de torque calculado |
| V | Função de Lyapunov |

LISTA DE SÍMBOLOS

| | |
|----------------|---|
| p_{sup} | Pressão se suprimento, Pa |
| u_i | Tensão de controle, V |
| p_{atm} | Pressão atmosférica, Pa |
| p_a | Pressão câmara a, Pa |
| p_b | Pressão câmara b, Pa |
| q_m | Vazão mássica, kg/s |
| A_i | Área do embolo, m ² |
| y_i | Posição do servoposicionador, m |
| \dot{y}_i | Velocidade do servoposicionador, m/s |
| \ddot{y}_i | Aceleração do servoposicionador, m/s ² |
| f_p | Força exercida pelo atuador, N |
| n | Numero de grau de liberdade |
| i | Variável de junta |
| ${}^{i-1}A_i$ | Matriz de transformação de um GDL |
| θ_i | Variável ângulo do elo da matriz D-H, rad |
| d_i | Variável de comprimento da matriz D-H, m |
| a_i | Distância entre elos da matriz D-H, m |
| α_i | ângulo entre elos da matriz D-H, rad |
| q | Variável de posição da junta |
| S | Superfície de deslizamento |
| φ | Camada limite |
| η | Velocidade na superfície de deslizamento, m/s |
| E/S | Entradas e saídas |
| ε | Erro de tolerância Newton-Raphson |
| τ | Vetor de torques ou forças do sistema, N.m |
| K_d | Matriz parâmetros do torque calculado |
| λ | Parâmetros torque calculado |
| τ_s | Torque desejado, N.m |
| $\tilde{\tau}$ | Erro de torque, N.m |
| s | Erro de seguimento posição e velocidade |

| | |
|-------|---------------------------------------|
| K_1 | Matriz de parâmetros de torque forças |
| p | Diferença de pressão nas câmaras, Pa |
| h_i | Intervalo de tempo entre pontos, s |

1 INTRODUÇÃO

A robótica é uma das áreas de pesquisa que vem apresentando elevados índices de crescimento nos últimos anos, principalmente devido à utilização de robôs em diferentes áreas, como, por exemplo, a indústria automotiva, medicina, mineração, serviços especiais de inteligência, entre outros. Para exemplificar, em 2009, após a crise financeira internacional, retomou-se a tendência de aumento na demanda por robôs manipuladores para usos industriais. Esta demanda deve-se, especialmente, à necessidade da diminuição dos custos da mão de obra, necessidade de alta qualidade de acabamento de produtos, eliminação de trabalhos repetitivos, de risco ou de elevado esforço [World Robotics, 2016].

Atualmente, uma das necessidades do setor industrial baseia-se na utilização de linhas de produção que, aplicadas em níveis organizacionais do chão de fábrica, devem ser competitivas e adaptáveis às mudanças do mercado. Considerando que as mudanças no ciclo de vida dos produtos são cada vez menores, os equipamentos devem ser facilmente reutilizáveis e reprogramáveis, além de atender às políticas de sustentabilidade ambiental [Cavalcante, 2012]. Para atender a estas necessidades, este trabalho propõe a utilização e implantação de módulos mecatrônicos na montagem de robôs modulares os quais, segundo Atta-Konadu et al., 2005; Larizza et al., 2006, poderão operar utilizando sistemas de controle distribuído, onde cada módulo consiste uma entidade autônoma, por meio de dispositivos conectáveis e auto configuráveis para a produção (segundo a ideia de *plug-and-produce* [Ferreira et al., 2012]). Os robôs com controle distribuído, quando montados, são formados por estruturas modulares, de forma que cada módulo possa ser facilmente substituído.

Há na literatura diversas abordagens que tratam da utilização de inteligência artificial em sistemas robóticos incorporando importantes avanços nos conceitos de mecatrônica, informática e eletroeletrônica aplicadas nesta área. Neste contexto, o presente trabalho objetiva propor uma estratégia de montagem e autoconfiguração de sistemas distribuídos, os quais serão utilizados na construção de robôs manipuladores de uso industrial. Neste caso, os componentes utilizados em um robô cilíndrico poderão, por exemplo, ser facilmente utilizados na montagem de um robô cartesiano, dentre outras possibilidades. No âmbito do presente estudo pretende-se elaborar um método genérico de projeto, desenvolvimento e implantação de um módulo mecatrônico a ser utilizado em aplicações de robótica industrial, baseado no conceito de arquiteturas automáticas híbridas (reativas e deliberativas) por meio de agentes [Wooldridge, 2009].

Para controlar os robôs modulares, os módulos mecatrônicos formam o núcleo científico/tecnológico do robô, sendo estes responsáveis pela realização da operação de controle, acionamento, sensoriamento e comunicação com outros robôs. Neste sentido, o presente trabalho visa a contribuir com a pesquisa na área de robótica aplicada, proposta em uma estratégia para o desenvolvimento de módulos mecatrônicos que poderão ser aplicados na montagem “rápida e simples” de robôs manipuladores.

Vale salientar que, com relação ao desenvolvimento e aplicações de arquiteturas modulares utilizadas na montagem de robôs manipuladores, existem poucas informações na bibliografia sobre as especificações gerais de plataformas eletrônicas, eletroeletrônicas, algoritmos de controle, estruturas mecânicas e dispositivos de acionamento. Além disso, documentações relacionadas à utilização de arquiteturas modulares na construção de robôs manipuladores com acionamento por meio de energia pneumática são escassas. A importância destes conceitos para a presente tese serão discutidos no desenvolvimento deste trabalho.

1.1 Motivação

A pesquisa aqui proposta apresenta-se como forma de consolidação e unificação dos ramos do conhecimento de desenvolvimento de controladores até então focados para servoposicionadores pneumáticos, e de procedimentos conceituais de arquiteturas robóticas modulares utilizados em sistemas mecânicos, ou, mais precisamente, em robô manipuladores. Como resultados espera-se disponibilizar uma estratégia para descrever e quantificar aspectos de operação, configurabilidade e customização necessários para a definição dos módulos mecatrônicos.

1.2 Descrição do problema

Nos últimos anos, os produtos manufaturados são cada vez mais caracterizados pela alta customização e oscilação de suas demandas, as quais obrigam as empresas a se adaptarem rapidamente para manter sua competitividade no mercado. Para esta finalidade, as linhas de produção devem reagir de forma conveniente, ou seja, as linhas produtivas devem ser capazes reorganizar os seus recursos de forma rápida e otimizada, em função das mudanças que possam existir na demanda.

A partir deste cenário, que visa a manter a produtividade e competitividade internacional das indústrias, pode-se destacar como temas de fundo:

1. É possível utilizar robôs modulares para reconfigurar diferentes arquiteturas de manipulação em uma linha de produção obsoleta?
2. Como configurar os robôs modulares de forma rápida em função das exigências de uma linha de produção?
3. É possível montar um robô manipulador de forma rápida e eficiente a partir de módulos básicos?
4. Sistemas de acionamento pneumático são viáveis para a montagem de robôs manipuladores modulares?

Fundamentado nas questões supramencionadas, é possível inferir que estudos relacionados à montagem de robôs manipuladores de uso industrial baseado em módulos mecatrônicos ainda apresentam questões que podem ser exploradas e adicionam as discussões sobre a utilização robôs modulares. Nesse sentido, destaca-se o fato de que ainda não existem normas estabelecidas ou procedimento formais para fabricação e montagens destes sistemas.

Em síntese, considerando os aspectos tratados, o presente trabalho propõe a definição e caracterização dos módulos mecatrônicos para a sua utilização na construção de robôs manipuladores, cujo gerenciamento da estrutura formada baseia-se em uma estrutura Multiagente. Para tanto, este trabalho proposto será validado por meio de um estudo de caso, mediante a utilização destes módulos mecatrônicos na montagem de um robô cartesiano de 3 GDL (graus de liberdade) de acionamento pneumático.

1.3 Metodologia

No desenvolvimento de novos produtos e no gerenciamento de projetos é adequado que sejam inicialmente realizadas a identificação e a subdivisão das áreas de conhecimento envolvidas no processo [Buzzetto, 2008]. Baseado nos procedimentos para o desenvolvimento de robôs propostos por Frasson, 2007 e aplicados por Sarmanho, 2014; e procedimentos para o desenvolvimento de robôs propostos por Chen e Yang, 1996 e Larizza et al., 2006; é proposta a seguinte metodologia para a execução do presente trabalho:

- **Projeto mecânico modular:** Definição dos requisitos mecânicos para a construção de um módulo mecatrônico com acionamento pneumático, baseando-se nos parâmetros de operação necessários para a utilização em robôs modulares com diferentes arquiteturas.
- **Composição eletromecânica e elétrica:** Realização de estudos associados à instalação dos dispositivos eletrônicos, eletroeletrônicos e eletropneumáticos

necessários para o funcionamento de um módulo mecatrônico. Para que estes dispositivos trabalhem de forma conjunta, é necessário atender a alguns requisitos específicos, os quais serão apresentados no desenvolvimento do trabalho.

- **Modelos matemáticos e identificação do sistema:** Descrição dos algoritmos matemáticos necessários para processar os códigos associados aos agentes mecatrônicos. Estes algoritmos incluem tarefas complexas como a utilizadas para a identificação cinemática do sistema recém configurado, sendo que estes algoritmos estão, portanto, encarregados de reconhecer a estrutura que foi formada, utilizando informações fornecidas por cada módulo acoplado ao manipulador. Em uma primeira abordagem, estes algoritmos deverão calcular os seguintes parâmetros: volume de trabalho, matrizes de cinemática inversa e direta, e o planejamento de trajetória para esta estrutura, como também a implementação de um algoritmo de controle desta estrutura formada.
- **Sistema baseados em agentes:** Configuração dos módulos mecatrônicos para operação segundo o conceito de agentes, os quais deverão possuir como características: trabalhar embarcado em cada módulo mecatrônico, determinar o comportamento de cada módulo por meio de suas habilidades e possuir a capacidade de interagir com outros módulos.

1.4 Objetivo do estudo

O principal objetivo deste trabalho consiste em estudar e definir os requisitos para a utilização de sistemas modulares na montagem de robôs manipuladores de uso industrial, atendendo às características de auto-organização da estrutura robótica formada baseadas no conceito de agentes de uso industrial.

Os objetivos específicos são os seguintes:

- a) Aplicação de módulos mecatrônicos na formação de manipuladores robóticos baseados em agentes;
- b) Especificação dos requisitos de projetos (mecânicos, eletrônicos e eletropneumáticos) de sistemas modulares, focando, especialmente, nos módulos mecatrônicos e suas configurações;
- c) Desenvolvimento de um algoritmo de identificação da arquitetura de um robô modular serial, determinando, assim, suas características cinemáticas;
- d) Implementação de algoritmos de controle local que serão aplicados a cada módulo;

- e) Implementação de algoritmos de geração de trajetórias globais que são utilizadas nos algoritmos de controle local de cada módulo mecatrônico;
- f) Apresentação um estudo de caso para a validação da proposta de tese mediante a simulação de um robô cartesiano modular.

1.5 Principais contribuições

A principal contribuição do presente trabalho é propor uma estratégia para a definição e montagem de robôs modulares, utilizando os conceitos de módulo mecatrônico e agentes. Para tanto, procura-se detalhar os requisitos de projeto e os componentes de um sistema modular e a descrição dos algoritmos computacionais que são processados para o funcionamento deste sistema. Além disso, é importante destacar que os procedimentos formais descritos, baseado em fundamentos teóricos, definem os requisitos de construção de módulos mecatrônicos, facilitando as ações dos fabricantes e usuários de robôs manipuladores, para que possam utilizar estes recursos em linhas de produção ou em tarefas customizadas.

1.6 Organização do trabalho

No Capítulo 2, é apresentada uma breve revisão a respeito de robótica, robôs modulares, sistemas baseados em agentes e módulos mecatrônicos, abordando os conceitos e suas importâncias no âmbito industrial. Seguidas da apresentação do estado da arte de robôs modulares de uso industrial. No Capítulo 3, é apresentada uma proposta conceitual para a formação de módulos mecatrônicos. Esta etapa baseia-se em apresentar os requisitos de projeto (mecânico e elétrico) de sistemas modulares, visando à formação de sistemas manipuladores. São também descritos os algoritmos matemáticos de formação e controle das estruturas, além da descrição da interação e aplicação dos agentes mecatrônicos. No Capítulo 4, é apresentado um estudo de caso de um robô modular cartesiano, onde é descrita a montagem, as configurações iniciais, o sistema organizacional, além de alguns aspectos de planejamento de trajetória deste sistema. No Capítulo 5, são apresentados resultados preliminares de simulações e discussões sobre o estudo apresentado. No Capítulo 6, são apresentadas as conclusões e sugestões para continuidade da pesquisa.

2 REVISÃO BIBLIOGRÁFICA

Para caracterizar um robô modular baseado em módulos mecatrônicos, é necessário abordar temas tais como (a) robótica industrial; (b) sistemas flexíveis para manufatura; (c) conceitos mecânicos de um robô; (d) módulos mecatrônicos; e (e) agentes e agentes mecatrônicos.

Estes conceitos, importantes para o embasamento teórico desta pesquisa, são apresentados nos capítulos que se seguem, juntamente com o estado da arte na área de desenvolvimento de robôs modulares.

2.1 Fundamentação teórica

Esta seção tem como objetivo apresentar os conceitos teóricos e definições necessárias para a definição de robôs modulares e módulos mecatrônicos, como será mostrado a seguir.

2.1.1 Estruturas cinemáticas de robôs clássicos

Os estudos em robótica caracterizam-se por integrar diversas áreas do conhecimento, como, por exemplo, a mecânica, a eletrônica, a inteligência artificial, a engenharia de controle e a física. Para Romano, 2002, um manipulador robótico é constituído, basicamente, por sistemas mecânicos compostos por elos, juntas e atuadores, os quais são utilizados para movimentação de ferramentas e peças, que, por meio de trajetória definidas, podem ser reprogramados de acordo com as necessidades. Dentre suas aplicações destacam-se tarefas que exigem alta repetitividade, precisão e segurança para os seres humanos, na sua utilização.

Atualmente, os robôs industriais ou manipuladores são os mais conhecidos e de maior utilidade nas linhas de produção, logo, na literatura, existem diversas definições para este manipulador que, em alguns casos, apresentam certa similaridade. Na Tabela 2.1, apresenta-se os conceitos abordados por diferentes organizações, os quais definem as características robôs industriais.

Tabela 2.1 – Conceitos de um Robô Industrial abordado por diferentes organizações

| Organização | Conceitos de um Robô Industrial |
|---------------|---|
| IFTToMM, 2015 | É uma máquina reprogramável para realizar funções de manipulação, movimentação e usinagem, possibilitando interagir com o ambiente e dotada de certo grau de autonomia. |
| BARA, 2016 | Consiste em um dispositivo programável e projetado para manipular peças, ferramentas ou implementos especializados de manufatura, permitindo de movimentos programáveis utilizados para a execução de tarefas específicas de manufatura. |
| RIA, 2016 | Tratam-se de um dispositivo multifuncional, reprogramável usado para realizar uma série de tarefas dedicadas à automação das atividades de um ambiente CIM (Manufatura Integrada por Computador). |
| ISO, 2012 | É um manipulador multifuncional reprogramável com vários graus de liberdade, capaz de manipular materiais, peças, ferramentas ou dispositivos especiais, seguindo trajetórias variáveis pré-programadas para realizar diversas tarefas. |
| AFNOR, 2016 | Abordam os robôs manipuladores em dois conceitos diferentes: Manipulador: mecanismos articulados entre si, destinado ao apreensão e deslocamento de objetos. Robô: manipulador reprogramável, capaz de posicionar e orientar peças, ferramentas ou outros dispositivos, seguindo uma trajetória reprogramável, para a execução de diversas tarefas. |

2.1.2 Conceito e tipo de agentes

Segundo Wooldridge, 2009, um agente é uma entidade informática autônoma que possui um conhecimento parcial do ambiente dinâmico circundante e tem a capacidade de interagir com outros agentes para alcançar um objetivo em comum. Em computação, um agente apresenta-se como um algoritmo racional ou inteligente, na qual existe um mecanismo intrínseco que define como o mesmo deve agir [Russell e Norvig, 2004].

Para a FIPA (*Foundation for Intelligent Physical Agents*), um agente é uma entidade que reside em um ambiente onde são interpretados os dados obtidos por meio de sensores, os quais percebem os eventos que ocorrem no ambiente, executando determinadas ações que produzem efeitos neste mesmo ambiente [FIPA, 2002].

Segundo Wooldridge, 2009, os agentes são definidos como um sistema computacional capaz de executar de forma autônoma ações em um ambiente no qual estes se situam, com o propósito de realizar objetivos a ele delegados. Normalmente, cada agente possui capacidades comportamentais, objetivos e poder de raciocínio necessários para sua interação em um meio ambiente. De maneira conceitual, um agente difere de um objeto, principalmente pelos

parâmetros que os definem (módulos, conjuntos de regras, base de conhecimentos) e pelo modo de comunicação com o exterior. Objetos realizam operações, por terem sido programados para tal, enquanto que agentes realizam operações porque decidiram realizá-la.

De acordo com Peixoto, 2012, um agente pode ser software puro ou hardware e um software composto. Cavalcante, 2012, complementa que um agente racional ou inteligente é aquele agente que é capaz de tomar uma decisão sobre como atuar a partir de conhecimentos prévios e do que o agente percebe do seu meio através de sensores.

De acordo com Hama, 2012, os agentes são entidades autônomas, pois são capazes de realizar observações através de sensores, agindo sobre um determinado ambiente, guiados por seus objetivos (baseado na seleção das informações de seu interesse). Além disso, estas entidades têm a capacidade de decidir quais ferramentas deve usar e, negociando com os demais agentes que habitam no sistema, quais são as ações mais cabíveis a serem tomadas em uma determinada situação. A Figura 2.1 apresenta um diagrama esquemático de um modelo de um agente simples.

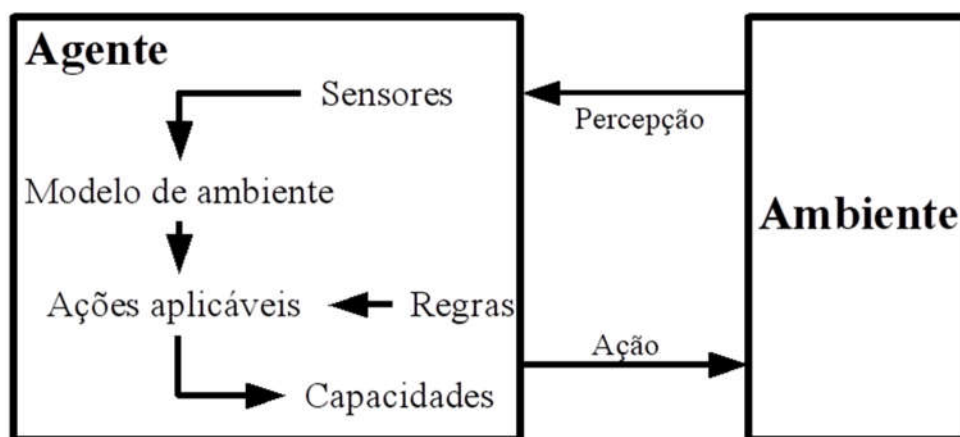


Figura 2.1 – Modelo de um agente simples [Hama, 2012].

A Figura 2.1 mostra que os agentes são sistemas que agem junto ao meio, pois são capazes de perceber as informações do ambiente por meio de sensores e de conhecimentos prévios (Modelo matemático), sendo as ações aplicadas pelo agente definidas a partir dos conhecimentos do meio e de regras preestabelecidas. As ações voltadas ao ambiente (ação), são definidas pelas ações aplicadas e limitadas pelas capacidades de atuação do próprio agente. Em síntese, pode-se dizer que um agente baseia-se nas capacidades de discernir como está o seu meio ambiente em um determinado momento, de realizar uma avaliação precisa de como e quando atuar para alcançar o seu objetivo [Posadas, 2004].

Neste trabalho, assume-se que um agente é definido como uma entidade (um software que é executado em um hardware apropriado) que pode perceber e/ou atuar em seu meio ambiente por meio de sensores e atuadores. Além disso, os agentes são definidos como entidades racionais ou inteligentes.

Sistemas Multiagentes (MAS)

O MAS ou Sistema Multiagente é uma tecnologia de software capaz de modelar e implementar comportamentos individuais e sociais em sistemas distribuídos. No conceito MAS, as entidades são chamadas de agentes ativos, pois são capazes de manifestar suas capacidades dentro de uma sociedade de agentes. Ademais, os agentes, em função de suas habilidades, submetem-se às solicitações de outros agentes, podendo assumir também um papel proativo para iniciar comunicação com outros agentes, propor negociação e alocar novas capacidades de processamento [Peixoto, 2012]. Dado que cada agente tem um conhecimento parcial do meio ambiente, o objetivo global é somente alcançado com a cooperação dentre os agentes que compõem a sociedade do MAS.

Os Sistemas Multiagentes podem ser classificados em três tipos: 1) MAS reativo, trabalham dentro de sistemas compostos por um grande número de agentes simples; 2) MAS cognitivos ou deliberativos, que, em geral, utilizam poucos agentes, mas que realizam tarefas mais complexas; e, por fim, 3) MAS híbridos, que têm propriedades tanto dos sistemas reativos como os deliberativos.

Segundo Posadas, 2004, o modelo MAS é caracterizado pelo uso de agentes que possuem uma certa autonomia. Os agentes podem interagir entre eles trocando informações, atuando em benefício de outros agentes, eventos internos ou externos executando funções autônomas.

Comunicação entre agentes e Sistemas Multiagentes

A *Foundation for Intelligent Physical Agents* (FIPA) define um conjunto de padrões para permitir a interoperabilidade entre os agentes físicos, independente da linguagem utilizada no seu desenvolvimento. Além disso, a FIPA estabelece vários protocolos e regras para comunicação entre agentes em vários níveis: aplicação, arquitetura, comunicação, gerenciamento de agentes e transporte de mensagens [FIPA, 2000, 2002; Poslad, 2007].

Dentre as especificações FIPA, destaca-se a *Agent Communication Language specifications* (ACL). Para esclarecer os conceitos sobre ACL, a seguir, são abordados o protocolo de interação de rede e a solicitação do protocolo de interação, ambos da FIPA, que correspondem ao mecanismo de negociação e execução de funcionalidades.

Protocolo de interação de rede (*Contract Net Interaction Protocol*)

De acordo com a FIPA, 2002b, este protocolo tem como objetivo permitir que vários agentes interajam. Neste caso, um agente iniciador solicita uma tarefa para os demais agentes do sistema (participantes), os quais são capazes de executar diferentes tipos de ações. Então, para atender à solicitação do agente iniciador, existe um processo de troca de informações com os demais agentes do sistema, definindo, assim, qual é o agente adequado para executar a tarefa desejada.

Há, neste protocolo, dois tipos de agentes na negociação: iniciador e participante. O agente iniciador é responsável por emitir uma mensagem *Call for Proposal* (CFP) para todos os agentes participantes do sistema. Os agentes participantes que sejam capazes de executar a ação solicitada responderão às mensagens com *Propose* (Disponível para a execução da tarefa) ou *Refuse* (indisponível para a execução da tarefa). Ainda, segundo a norma, um agente que tem a capacidade de executar o serviço solicitado, mas está ocupado no momento em que recebe o CFP, envia uma mensagem de *Refuse* por meio da rede de comunicações.

Após o agente iniciador encerrar a chamada de solicitação, ele analisa as propostas recebidas, escolhendo, por meio de algum critério definido, qual ou quais agentes deverão executar a atividade desejada.

Quando as propostas são aceitas pelo agente iniciador, são enviadas para a rede as mensagens *Accept Proposal*, enquanto que, para as propostas não aceitas, são enviadas para a rede as mensagens *Reject Proposal*. Quando a proposta é aceita pelo agente iniciador, este fica no aguardo de duas possíveis ocorrências (mensagens): A mensagem *Inform* consiste em finalizar corretamente a execução da tarefa solicitada pelo agente iniciador. A mensagem *Failure* informa que a tarefa não foi realizada ou ocorreu uma falha por parte do agente executor. A Figura 2.2 apresenta um diagrama sequencial do protocolo *FIPA Contract Net Interaction Protocol* [FIPA, 2002]..

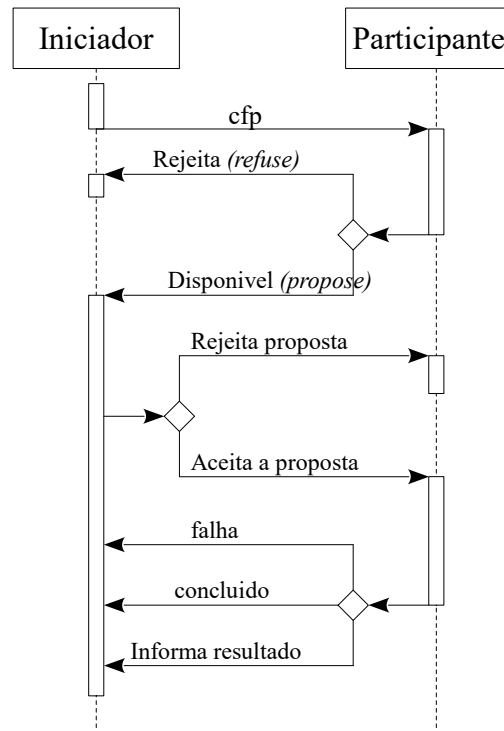


Figura 2.2 – FIPA *Contract Net Interaction Protocol*.

Este protocolo, analogamente, trata-se de uma arquitetura do tipo cliente/servidor convencional. Nesta abordagem, quando dois agentes interatuam, uma mensagem é enviada pelo agente iniciador (cliente) para uma entidade específica (servidor), requisitando uma tarefa a ser realizada. O servidor, por sua vez, é responsável pelo processamento da solicitação, retornando a resposta solicitada.

Solicitação de protocolo de interação (*Request Interaction Protocol*)

É importante destacar que, neste protocolo, a solicitação realizada pelo agente iniciador, pode ser ou não atendida no momento em que for requisitada. Desta forma, se o pedido de solicitação é aceito, o participante informa o iniciador que irá realizar o serviço requisitado (mensagem *Agree*). No entanto, se o pedido de solicitação não é aceito, existem duas mensagens de retorno possíveis. Se o serviço é realizado, é enviado de volta ao iniciador uma mensagem *Inform*, caso contrário, é enviada uma mensagem *Failure*.

A Figura 2.3 apresenta um diagrama sequencial do protocolo FIPA *Contract Net Interaction Protocol*.

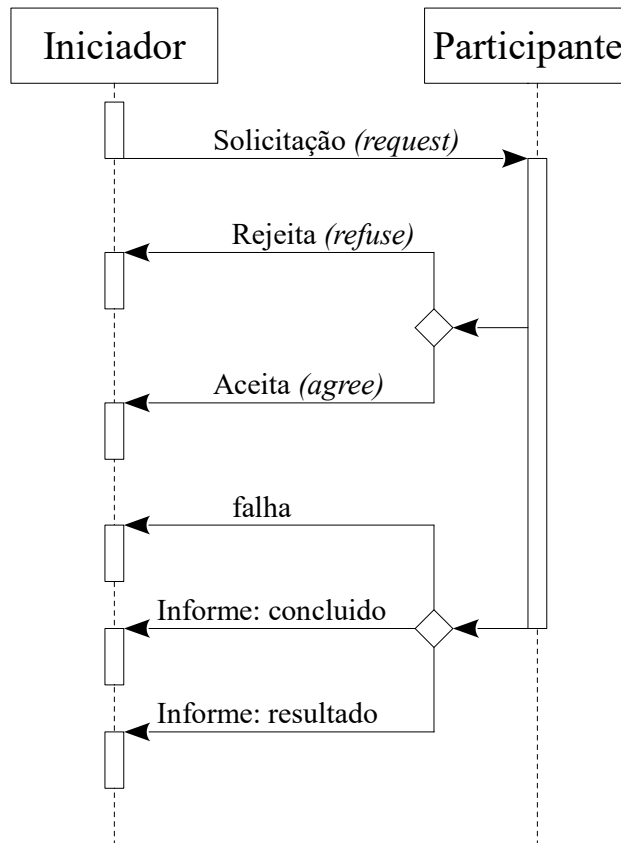


Figura 2.3 – FIPA *Request Interaction Protocol* [FIPA, 2002].

Conteúdo das mensagens FIPA

As mensagens FIPA, que consistem em pacotes de dados, são formadas por diversos campos, tais como:

- **Destinatário:** consiste na identificação do agente no sistema AID (*Agent IDentifier*);
- **Conteúdo:** representam objetos em série por meio de uma sequência de caracteres;
- **Controle das mensagens:** trata-se da identificação de conversas, do tipo de mensagem, da temporização máxima, entre outros;
- **Ontologia:** é um campo opcional, usado para enfatizar uma mensagem dentro de um certo contexto de trocas de mensagens;
- **Parâmetros da mensagem:** permitem ao usuário personalizar cada mensagem;

É importante destacar que o conteúdo das mensagens associado aos campos de ontologia e parâmetros das mensagens permitem a comunicação com uma aplicação em mais alto nível dentro do próprio protocolo.

2.1.3 Módulo Mecatrônico e Agente Mecatrônico

Para melhor compreensão dos temas apresentados nesse tópico, é inicialmente necessário realizar uma breve discussão sobre os conceitos de módulos e suas particularidades.

Um módulo é uma unidade autocontida que possui uma função específica, uma interface bem definida e a possibilidade de interagir com outros módulos [Cavalcante, 2012; Posadas, 2004]. Por outro lado, um sistema mecatrônico pode ser projetado para ser modular, ou seja, para que tenha uma funcionalidade e interfaces bem definidas, tanto mecânicas, quanto elétricas e/ou computacionais. Com isso, pode-se definir um módulo mecatrônico como um módulo formado por partes mecânicas, eletrônicas e de comunicação de software integrados, com capacidade de percepção e atuação física [Cavalcante, 2012; Larizza et al., 2006; Posadas, 2004].

Um agente mecatrônico é um agente racional ou inteligente contido em um módulo mecatrônico, ou seja, é capaz de perceber, decidir quando e como agir, e atuar no mundo físico, autonomamente. Um agente mecatrônico é uma entidade, geralmente um *software* executado em um *hardware* apropriado, que pode perceber e atuar em seu meio ambiente [Cavalcante, 2012; Ribeiro et al., 2011]. Assim, no presente trabalho, ao citar-se agentes, é subentendido que os mesmos são agentes racionais e inteligentes, exceto se for explicitamente indicado o contrário.

Para fins práticos, neste trabalho, um agente mecatrônico será identificado como um módulo mecatrônico que o executa, pois há muita semelhança entre eles, e sempre um agente mecatrônico estará associado a um módulo mecatrônico. É pouco comum a existência de um módulo mecatrônico associado a mais de um agente racional, mesmo se uma abordagem MAS for utilizada no seu desenvolvimento. A Figura 2.4 ilustra a relação entre os módulos mecatrônicos e os fatores externos necessários para o seu funcionamento.

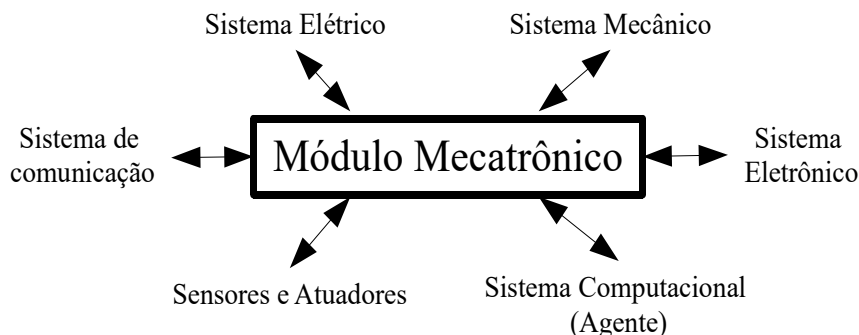


Figura 2.4 – Interação entre sistemas dentro de um módulo mecatrônico

Características construtivas dos Módulo Mecatrônicos

Esta seção apresenta os conceitos básicos relacionados à metodologia de projeto de um produto de uso industrial, visando a, posteriormente, projetar módulos mecatrônicos, os quais são utilizados na formação e montagem de robôs modulares. A metodologia descrita baseia-se nos conceitos apresentados por [Pahl et al., 2007].

As fases de desenvolvimento do projeto são:

- **Especificação de Projeto:** Nesta fase são coletadas as informações pertinentes ao produto a ser desenvolvido, e, com estes dados compilados, são definidos os requisitos do projeto.
- **Projeto Conceitual:** Nesta etapa devem ser consolidados os conceitos tecnológicos funcionais e morfológicos de um determinado produto de uso industrial. Também é efetuada a seleção de soluções.
- **Projeto Preliminar:** Nesta fase são determinadas as informações técnicas, referentes ao produto em estudo.
- **Projeto Detalhado:** Nesta etapa é produzida a documentação para a produção industrial e a avaliação deste determinado produto.

Neste trabalho, para descrever o projeto de um módulo mecatrônico, serão abordadas as etapas associadas à especificação do projeto, projeto conceitual e projeto preliminar.

Sistema elétricos de um módulo mecatrônico

Um módulo mecatrônico possui uma arquitetura elétrica concebida para atuar de forma flexível, sendo o controle individual de cada unidade efetuado em um hardware dedicado. A ideia de separação ou distribuição do sistema de controle em unidades já foi aplicada com sucesso em manipuladores robóticos com acionamento elétrico [Santini, 2009]. Uma proposta semelhante, porém, aplicada a servoposicionadores pneumáticos, pode ser encontrada em Cukla, 2012.

Esta arquitetura baseia-se em uma plataforma de processamento de dados embarcados em um sistema operacional, que, utilizando dispositivos elétricos e eletrônicos, trabalha na aquisição de dados (pressão, posição, aceleração, etc.) e atuação (servoposicionador pneumático). Os principais componentes eletrônicos de um módulo mecatrônico estão dedicados ao processamento de dados e comunicação do sistema, embora existam componentes

que realizam o tratamento de sinais analógicos (condicionamento de sinais de entrada e saída do sistema de processamento).

A seguir, são apresentados os conceitos teóricos relacionados às diferentes partes elétricas que compõem o módulo mecatrônico.

Microcontrolador

O microcontrolador é um circuito integrado com a capacidade de execução de funções (processamento de dados) através de instruções específicas (softwares), sendo constituído por um microprocessador e um ou mais periféricos, tais como: memória, temporizador/contador, Conversores Digital/Analógico ou Analógico/Digital, comunicação (série, paralelo, CAN, etc.), entre outros.

Os microcontroladores são componentes que, além de suportar sistemas de controle lógico, realizam tarefas específicas, isto é, um microcontrolador é programado para cumprir uma determinada função, interagindo com o meio. Tais componentes são escolhidos para este tipo de atividades, pois são compactos, versáteis e autossuficientes na maioria das suas operacionalidades [Intel, 2016; Microchip, 2016]. Estes circuitos integrados apresentam bom desempenho para o tratamento e aquisição de sinais, além de sua interface com o meio físico através de seus periféricos. No entanto, os microcontroladores não possuem a capacidade de processar algoritmos sofisticados, como também não são flexíveis na comunicação com sistemas comerciais, porque sua implementação pode necessita de maior trabalho no desenvolvimento dos algoritmos computacionais.

Computador principal

Conforme já comentado, considerando o desempenho dos controladores disponíveis nos dias de hoje, o processamento dos algoritmos de controle de posicionamento, a comunicação com outros módulos, a execução dos agentes e o processamento de dados em tempo real devem ser realizados por um sistema computacional de maior capacidade do que um microcontrolador. Assim, a plataforma selecionada deve suportar diferentes linguagens de programação, compatibilidade com *softwar* e com sistemas operacionais de código aberto, facilitando e viabilizando o desenvolvimento de trabalhos futuros. Além disso, o computador principal deve ser de tamanho reduzido, de baixo consumo de energia, e dispor de periféricos de entrada e saídas compatíveis com os de uso comerciais (USB, HDMI, Ethernet, entre outros).

O Raspberry Pi®, que consiste em um computador principal (Raspberry Pi modelo B) preliminarmente avaliado no âmbito do presente trabalho no sistema elétrico de um módulo mecatrônico, é um computador desenvolvido no Reino Unido pela Fundação Raspberry Pi. Neste caso, o hardware é integrado em uma única placa e seu objetivo principal é estimular o ensino de ciência da computação em unidades acadêmicas [Raspberry, 2016].

O Raspberry Pi é um computador de baixo custo, de dimensões aproximadas às de um cartão de crédito e pode ser acessado por um monitor de computador ou TV (teclado e mouse padrão) e/ou por rede através de outro computador. Este sistema pode ser programado em C, C++, Python e ainda suportar ferramentas matemáticas para engenharia, tais como o Octave e o Scilab. O Raspberry Pi tem capacidade semelhante às de um computador *desktop*, permitindo navegar na Internet, gerar planilhas, realizar processamento de texto, implementar modelos e simular comportamentos de sistemas.

Entre outras vantagens, o Raspberry Pi tem a capacidade de interagir com o ambiente externo, podendo ser utilizado em uma ampla gama de projetos de fabricação de dispositivos digitais, máquinas multimídia, gestores de estações meteorológicas, sistemas de processamento de imagens térmicas baseadas em câmaras com sensores infravermelho, entre outros. Para que este computador possa interagir com o mundo exterior (meio físico), o Raspberry Pi contém 26 pinos de uso geral, os quais podem ser usados para a comunicação com outros dispositivos [Raspberry, 2016].

Vale ressaltar que, apesar de não ser objeto desta pesquisa, existem outros modelos comerciais de microcomputadores semelhantes ao Raspberry Pi, tais como, o *BeagleBone Black*, o *HummingBoard*, *Banana Pi* e *Odroid U3*.

Sistema de comunicação

Os módulos mecatrônicos propostos no presente trabalho, dentre suas diversas funcionalidades, caracterizam-se pelo alto fluxo de informações entre o microcontrolador e o microcomputador. Desta forma, para estabelecer a comunicação entre os componentes, propõe-se a utilização de um protocolo de comunicação serial síncrona, o qual visa a facilitar o intercâmbio de dados e o controle em tempo real em um ciclo de curta duração. Este, preferencialmente, deve ser menor do que 1ms, valor geralmente são utilizados em sistemas pneumáticos de posicionamento (conforme descrito no ANEXO A – ALGORITMOS DE CONTROLE).

Alguns dos protocolos de comunicação mais populares em ambientes industriais são: RS-232, RS-485, CAN, Ethernet e o EtherCAT, entre outros. Para estabelecer a comunicação entre o microcontrolador e o cilindro pneumático e, conseqüentemente, transmitir os dados de posição do pistão, o presente trabalho propõe a implementação dos protocolos de comunicações CAN e SPI (*Serial Peripheral Interface*). Estes protocolos também são utilizados na aquisição de dados, tais como a pressão das câmeras dos cilindros pneumático e a transmissão de dados entre o microcomputador e o conversor analógico digital.

2.1.4 Componentes pneumáticos e eletropneumáticos

O sistema mecânico acoplado a cada módulo mecatrônico é composto pelos seguintes elementos: um atuador linear ou rotacional de dupla ação, uma servoválvula proporcional controlada, o sistema eletrônico, os sensores de pressão e aceleração, os vasos de pressão e o filtro de ar. Seguem-se as descrições gerais desses componentes, nos próximos subtópicos.

Atuadores pneumáticos de acionamento linear

São também conhecidos comumente como cilindros pneumáticos, são elementos constituídos por um cilindro linear, onde em uma de suas extremidades contém a conexão que serve de admissão e exaustão do ar, e na outra extremidade, outra tampa com iguais características, porém, normalmente, dotada de um furo central onde se movimenta uma haste que, na extremidade interna ao cilindro, possui um êmbolo com vedação, que, pela ação do ar expandindo-se no interior do tubo cilíndrico, possibilita o movimento de expansão ou retração dessa haste (para os casos de cilindro com haste) [Fialho, 2004]. Os atuadores pneumáticos são regidos pelas normas internacionais ISO 6431 e ISO 6432. Um cilindro pneumático sem haste transmite potência sem necessidade de movimentar um elemento que precise aumentar o espaço ocupado pelo atuador, sendo esta uma característica apropriada para sistemas modulares [FESTO, 2016].

Servoválvulas de comando

São dispositivos eletropneumáticos que, a partir de um impulso elétrico, permitem a passagem de fluxo de ar pressurizado para alimentar dispositivos de acionamento pneumático

sistema tem como entrada uma tensão u_i e como saída o deslocamento y do êmbolo do cilindro. Esta dinâmica compreende:

- a relação entre a tensão aplicada u_i e o deslocamento do carretel (dinâmica da válvula);
- as relações entre as aberturas na válvula (ocasionadas pelo deslocamento do carretel) e as vazões mássicas de ar q_{ma} e q_{mb} que atravessam os orifícios de controle da válvula;
- as relações entre as vazões mássicas nas câmaras do cilindro e as pressões p_a e p_b que elas provocam;
- a relação entre a diferença de pressões e a área do êmbolo com a força f_p resultante.

Para Sobczyk, 2009, um servoposicionador é um elemento que possibilita o posicionamento de uma carga mecânica em uma localização desejada. Esta localização é usualmente denotada pelos eixos coordenados x , y ou z , podendo ser fixa ou variável no tempo. Normalmente, estes sistemas necessitam de um controlador capaz de operar em malha fechada, sendo a variável de controle a posição do elemento de trabalho.

Os servoposicionadores pneumáticos têm-se mostrado eficazes em diversas aplicações onde são necessários movimentos lineares, como, por exemplo, o transporte de objetos a curtas distâncias, operações de fixação e prensagem leve [Fialho, 2004].

2.1.5 Sistema robótico modular

De acordo com Chen e Yang, 1996 para a junção modular do sistema mecânico (sistema robótico modular), é necessário levar em consideração os seguintes conceitos:

- **Homogeneidade:** um módulo deve ser homogêneo em hardware e software, pois os módulos devem ser intercambiáveis e capazes de substituir partes do sistema que esteja defeituoso.
- **Controle local:** o comportamento da unidade mecatrônica é determinada pelas informações locais do sistema, permitindo que o sistema possa agir de forma distribuída sem coordenação centralizada.
- **Reconfiguração dinâmica:** os módulos, de acordo com sua localização dentro do sistema, devem ser capazes de mudar sua configuração automaticamente. Essa característica é especialmente importante para garantir a automontagem de sistemas robotizados.

Um robô reconfigurável pertence a um tipo de sistema robótico que tem a capacidade de mudar a sua forma geométrica e funcionalidades, utilizando, para qualquer arquitetura robótica, os mesmos módulos básicos [Murata et al., 2002]. Cada módulo é equipado com um sistema computacional responsável por realizar a comunicação, a leitura de sensores e o comando dos atuadores. Nestes robôs, os módulos mecatrônicos têm a capacidade de ser conectados de diferentes formas, ou seja, o sistema pode ser reconfigurado em diferentes estruturas e funcionalidade. Quanto às possíveis estruturas dos robôs modulares, é possível obter configurações do tipo antropomorfo, cilíndrico, cartesiano, esférico, etc.

Estes robôs, por serem flexíveis, tornam-se adequados a ambientes industriais controlados, como, por exemplo, o chão de fábrica de uma planta industrial, onde um mesmo robô de origem modular pode ser adaptado a diferentes necessidades da linha de produção [Chen e Yang, 1996; Larizza et al., 2006].

Entre as vantagens deste sistema, destaca-se a sua rápida reconfiguração, a qual é devida ao fato de que estes se caracterizam por disporem de formas predeterminadas de conexão. Além disso, também contribui o fato de que o processamento dos algoritmos de controle de posicionamento é realizado localmente (dentro de cada módulo mecatrônico) e relativo a cada unidade. No entanto, alguns requisitos devem ser impostos ao hardware modular, devido às limitações de volume, à necessidade de conexões (elétricas, mecânicas e pneumáticas) confiáveis e a dificuldades associadas à movimentação relativa de cada módulo.

No presente trabalho, os módulos propostos são constituídos por uma ligação mecânica, os quais são fixados junto aos módulos auxiliares, possibilitando a conexão de diferentes formas heterogêneas modulares (dois tipos de módulos).

Os módulos mecatrônicos básicos podem ser definidos em dois tipos: de acionamento prismático ou linear e de acionamento rotacional. Como definição no âmbito do presente trabalho, um módulo mecatrônico de acionamento pneumático, na grande maioria das abordagens, refere-se a um módulo de acionamento linear. Assim, para que um módulo com acionamento rotacional seja considerado na montagem de um robô, é necessário especificá-lo no problema, pois, caso contrário, considera-se que se está adotando módulo de acionamento linear.

Componentes básicos de um robô modular

Segundo Chen e Yang, 1996; Larizza et al., 2006, os módulos utilizados em robôs modulares podem ser definidos em dois tipos: ativos e passivos. Os módulos ativos consistem

em sistemas mecânicos que dispõem no seu interior de dispositivos eletrônicos que realizam o controle ou processamento de dados (estes são denominados módulos mecatrônicos). Já, os módulos passivos são sistemas que apresentam um elemento de união entre dois módulos mecatrônicos (módulos auxiliares).

Considera-se que os módulos mecatrônicos consistem de módulos ativos, pois apresentam características de autonomia, podendo funcionar de maneira isolada em uma dada função elementar. Quando se trata de um sistema modular, que é composto por um conjunto de módulos mecatrônicos, a coalisão entre eles é obtida por meio de um módulo auxiliar ou elo mecânico, o qual permite a troca de movimentos relativos e esforços mecânicos entre as diferentes partes do sistema.

Na Figura 2.6, apresenta-se um diagrama de blocos que representa um módulo ativo básico no acionamento de atuadores prismáticos [Larizza et al., 2006].

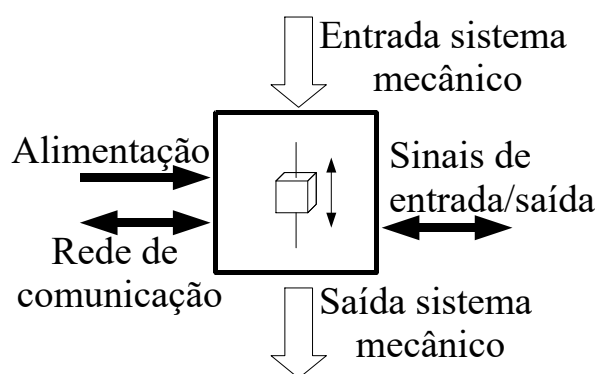


Figura 2.6 – Módulo ativo básico.

Os robôs modulares são construídos por meio de módulos independentes que devem ser compatíveis entre si. Desta forma, as unidades básicas (módulos mecatrônicos) são acopladas através dos módulos passivos ou módulos auxiliares. A união entre as unidades básicas e os módulos auxiliares permitem a transmissão de energia (elétrica e pneumática), movimentos e esforços mecânicos entre os demais módulos mecatrônicos que constituem o sistema. Na Figura 2.7, apresenta-se um exemplo de possível configuração modular para robôs SCARA [Larizza et al., 2006].

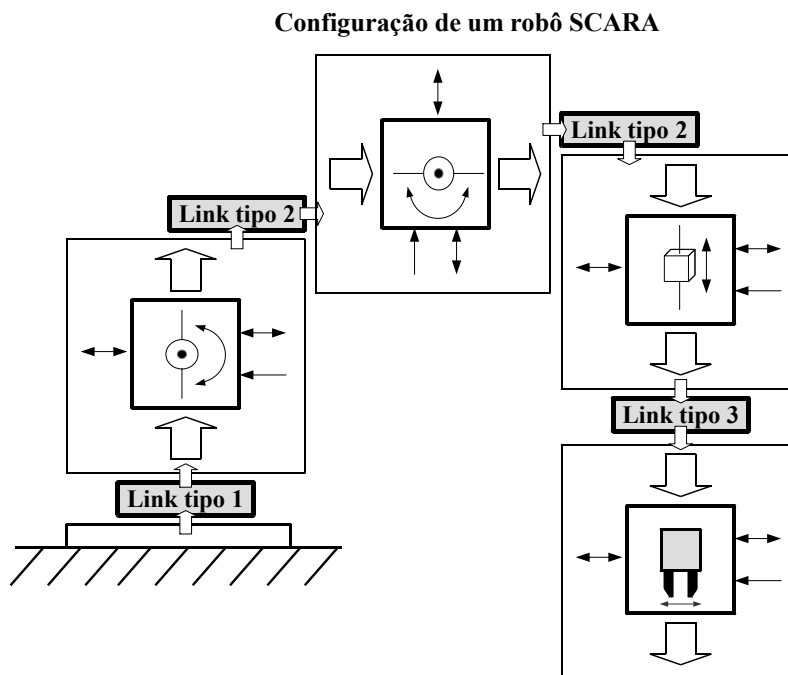


Figura 2.7 – Configuração modular de um robô SCARA.

Neste trabalho, no caso de um robô modular, assume-se que o tipo de ligação entre os módulos mecatrônicos possibilita a identificação das posições, do tipo de acionamento dos mesmos e das características cinemáticas e dinâmicas globais por meio do sistema de gerenciamento. Estas particularidades permitem determinar as características globais do sistema montado, bem como definir as diretrizes do controle, planejamento de trajetórias e, conseqüentemente, as tarefas específicas do robô (funções *pick-and-place*, paletização, etc.).

2.2 Estado da Arte

Nesta seção, são descritos os trabalhos encontrados na literatura que tratam sobre a utilização de robôs modulares, os sistemas industriais baseados em controle distribuído e a aplicação de algoritmos evolutivos baseados em agentes mecatrônicos. É importante destacar que bibliografias sobre a utilização de robôs modulares baseados em agentes, particularmente para sistemas que utilizam energia pneumática como meio de acionamento, são escassas.

Para caracterizar o estado da arte sobre robôs manipuladores utilizando módulos mecatrônicos, serão descritos os aspectos considerados relevantes, bem como as principais pesquisas utilizadas no desenvolvimento deste trabalho.

(i) Tipo de acionamento das juntas de um robô modular: Aqui busca-se referenciar trabalhos que tratam sobre a montagem de robôs modulares, independentemente do tipo de

energia de acionamento das juntas. Dentre os diversos tipos de energia utilizados no acionamento das juntas, pode-se mencionar os de origem elétrica, hidráulica e, em menores proporções, pneumática [Sarmanho, 2014].

Segundo Chen e Yang, 1996, um robô modular consiste na padronização de unidades de acoplamento (módulos passivos) e juntas (módulos ativos), possibilitando, assim, a montagem de diferentes configurações cinemáticas de robôs industriais. No caso de robôs modulares, um módulo, em uma mesma estrutura física, é composto por um elo, uma junta (rotacional ou prismática), uma unidade de processamento de dados e uma unidade de comunicação em uma mesma estrutura física. Além disso, os módulos utilizam os mesmos tipos de energia de acionamento que os robôs industriais.

Em síntese, com relação ao tipo de acionamento das juntas de um robô modular, este trabalho procura analisar, dentre os trabalhos encontrados na literatura, as seguintes características: a estrutura do controlador, a autonomia das unidades de acionamento e as fontes de energia utilizadas no acionamento dos módulos, conforme apresenta-se a seguir.

(ii) Sistemas de uso industrial baseados em agentes: Este item procura analisar as abordagens que, em sistemas de uso industrial, utilizam softwares de gerenciamento baseados em agentes e realizam uma distribuição de suas atividades.

Na bibliografia pesquisada, observou-se que os diversos autores analisaram e utilizaram os sistemas industriais baseados em agentes de diferentes maneiras. Estudam aplicações em processos de manufatura, ou abordam problemas em robôs móveis, como apresentado a seguir.

(iii) Sistemas mecatrônicos de uso industrial: Como já dito, os sistemas mecatrônicos de uso industrial são constituídos por componentes mecânicos, elétricos, eletrônicos e computacionais, de forma que a integração destes elementos compõem uma plataforma de uso industrial, como, por exemplo, os AGV, as estações de trabalho, entre outros [Peixoto, 2012; Rocha, 2013]. Outra característica destes sistemas é que são capazes de processar algoritmos computacionais mais complexos, como os agentes.

Desta forma, o presente item objetiva apresentar os avanços e as implementações de sistemas que utilizam módulos mecatrônicos em aplicações industriais. Tais aplicações referem-se, principalmente, à robótica e aos sistemas autônomos aplicados às linhas de produção.

(iv) Sistemas de controle distribuído: Segundo Larizza et al., 2006, os sistemas de controle distribuído são mais comumente utilizados em processos industriais, apresentando várias vantagens como, maior robustez, segurança operacional, escalabilidade, entre outras. Atta-Konadu et al., 2005, complementa que, nestes sistemas, os algoritmos de controle

distribuído podem apresentar menor complexidade na implementação de cada módulo mecatrônico, porém, na maioria de suas aplicações, requerem uma alta taxa de troca de informações.

O presente item procura analisar as pesquisas que apresentam propostas de sistemas robotizados ou outras aplicações industriais, analisando e descrevendo, brevemente, as estratégias de controle distribuído que foram utilizadas em tais aplicações.

2.2.1 Robôs modulares e agentes mecatrônicos

Em Chen e Yang, 1996, foi proposto um robô modular de uso industrial. O objetivo desta pesquisa foi desenvolver um sistema de pequenas dimensões capaz de trabalhar nas linhas de montagem de pequenas peças, como é a indústria eletrônica. Neste caso, o estudo consiste em uma proposta onde é realizada a montagem e definição matemática do modelo cinemático do robô modular configurado. Nesta abordagem, o autor apresenta diferentes configurações de estruturas seriais, a partir de unidades básicas de acionamento linear ou rotacional, onde foram realizadas montagens de estruturas de até 3 graus de liberdade.

O sistema proposto por Chen e Yang, 1996, é baseado em blocos de formatos cúbicos, onde estão alojados os sensores, o sistema eletrônico de condicionamento e a unidade de acionamento. Estes cubos, quando montados de forma sequencial, formam uma estrutura serial articulada que, de acordo com as características dessa estrutura, define a configuração de um robô modular de uso industrial. Construtivamente, os blocos têm duas formas específicas de conexão já preestabelecidas nos módulos, os quais podem transferir energia e movimentos mecânicos para outros módulos ou, até mesmo, para o efetuador. Os blocos funcionais possuem acionamento linear ou rotacional. O movimento destes componentes é realizado por um sistema de transmissão e pequenos motores elétricos de passo. O sistema de controle do robô modular é de arquitetura centralizada, e não se encontra alocado dentro de um módulo. Cada unidade modular não possui um hardware adequado para a realização de processamento de dados e controle local. A Figura 2.8 apresenta dois tipos de configurações propostas pelo autor [Chen e Yang, 1996].

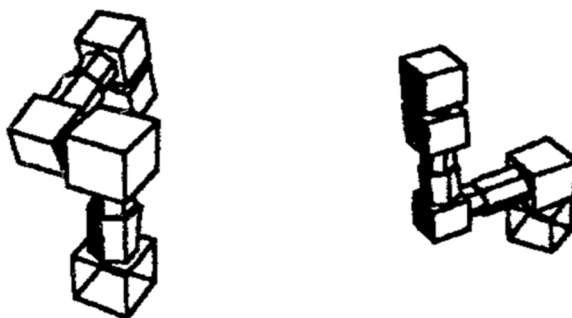


Figura 2.8 – Dois tipos de configurações de robôs modulares.

É importante destacar que esta abordagem não apresenta nenhum tipo de algoritmo especializado e/ou estratégias de aprendizagem do meio ambiente para melhorar o desempenho do sistema, ou seja, em outras palavras, este sistema não utiliza agentes como algoritmo de gerenciamento destes módulos. Ademais, este trabalho detalha como são calculadas as matrizes cinemáticas através dos parâmetros de Denavit-Hartenberg, que na abordagem proposta pelo autor são inseridos manualmente, de acordo com a estrutura que se deseja montar. O controle centralizado, o qual realiza cálculos mediante métodos iterativos, determina a posição das juntas e do efetuator em função da cinemática direta e inversa.

Apesar deste trabalho tratar-se de um grande avanço na área de robótica modular de uso industrial, pois aborda o conceito de modularidade de sistemas mecânicos (por exemplo, manipuladores), nesta abordagem, os módulos apresentados pelo autor não podem ser definidos como módulos mecatrônicos, uma vez que não possuem uma plataforma de processamento de dados, como um microcomputador ou um microcontrolador com capacidade de processamento de algoritmos sofisticados. Além disso, não apresenta uma abordagem conceitual sobre os algoritmos especializados para controle de posição, tais como, os sistemas de controle distribuído.

Atta-Konadu et al., 2005, apresenta uma arquitetura elétrica e de controle distribuído para robôs industriais reconfiguráveis, onde cada módulo elétrico processa, em tempo real, os algoritmos de controle local, possibilitando a escalabilidade dos módulos do sistema. Para cada controlador, propõe-se um sistema que trabalha em rede, na qual cada unidade de acionamento é constituída de uma unidade de processamento (microprocessador aJ-100 compatível com Java). Nesta estrutura, os algoritmos de controle são processados em forma distribuída, bem como a gestão de trajetória, a identificação dos módulos e a sincronização dos dispositivos. Em relação ao algoritmo de controle de cada unidade desenvolvida, o trabalho não faz uma análise detalhada dos tipos de algoritmos utilizados para cada eixo ou módulos dos sistemas.

A Figura 2.9 ilustra um diagrama simplificado desta arquitetura que é aplicada ao controle de robôs (podendo serem estes de uso industrial) de acionamento elétrico.

Nesta pesquisa, a arquitetura elétrica do robô modular, apresenta características similares à de um sistema evolucionável, como, por exemplo, a escalabilidade, a reconfigurabilidade e o uso de código aberto. Apesar de abordar tais características, utilizam-se os sistemas evolucionáveis e, consequentemente, os agentes na implementação de unidades elétricas controladoras utilizadas em robôs modulares.

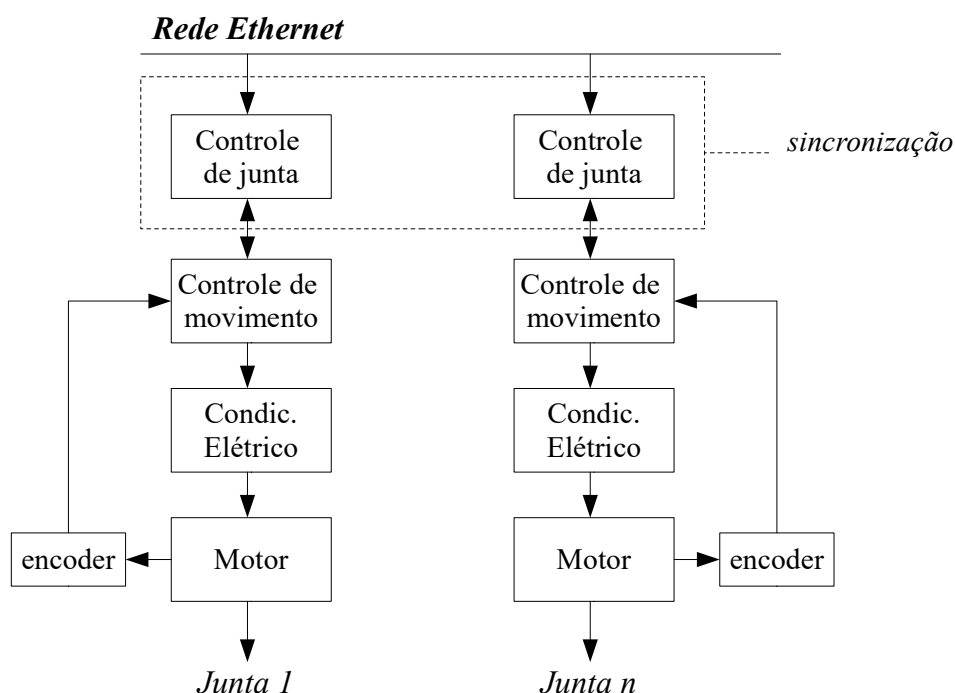


Figura 2.9 – Arquitetura descentralizada proposta por Atta-Konadu et al., 2005.

Para atender os requisitos de escalabilidade, cada unidade apresenta determinadas características, tais como, fácil inserção e retirada de unidades do sistema, autonomia no processamento de dados e atuação, entre outras. Para tanto, o autor propõe a utilização do protocolo *ZeroConfig*, que é dedicado à inspeção e detecção de todas as unidades no sistema dentro de uma rede Ethernet. A geração de trajetória e os cálculos cinemáticos são realizados pela unidade supervisora do sistema, que é uma unidade externa ao robô. Esta unidade é responsável por identificar e delegar as tarefas para as outras unidades do sistema (*multicast sockets*). No processamento dos ciclos de controle, as unidades informam para o sistema principal quando as atividades de cada unidade foram realizadas, habilitando, assim, a

supervisão e sincronização das entidades. A Figura 2.10 apresenta a sequência de comandos entre as unidades de controle e o supervisor [Atta-Konadu et al., 2005].

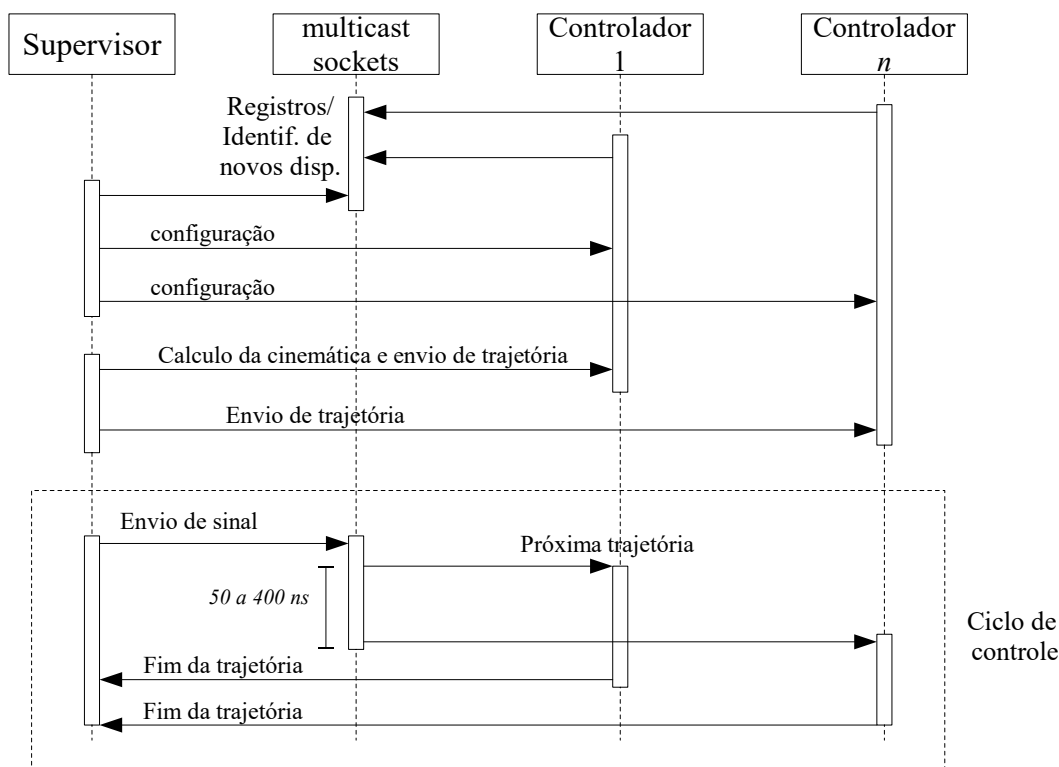


Figura 2.10 – Diagrama de sincronização das unidades de controle.

Ainda, segundo o autor, embora esta pesquisa tenha especificado as características de uma plataforma eletrônica para um robô de uso industrial e controle distribuído, existem conceitos que não foram abordados em relação à construção e à implementação das unidades, como o conceito de módulos mecatrônicos. É importante destacar que este trabalho trata de uma arquitetura de sistema de controle distribuído, no qual dispõe de um supervisor encarregado pela interface com usuário, pelo planejamento de trajetória e pela distribuição de tarefas para cada unidade. Além disso, nesta abordagem, as atividades de comunicação entre as unidades controladoras das juntas do robô não são realizadas em tempo real.

Larizza et al., 2006, propõe a construção de uma nova geração de robôs modulares de uso industrial, os quais apresentam estrutura distribuída e unidades totalmente autônomas. Estes manipuladores, que são utilizados em aplicações de linhas de montagens industriais, caracterizam-se por formar diferentes estruturas cinemáticas, tais como cadeias cinemáticas seriais, paralelas ou combinadas. Nesse trabalho, para definir um módulo mecatrônico, são tratados os seguintes aspectos: a utilização de sistemas eletromecânicos de atuação, o circuito

eletrônico de leitura de sensores e atuação, o sistema de processamento de dados e, por fim, o sistema de comunicação de alta eficiência.

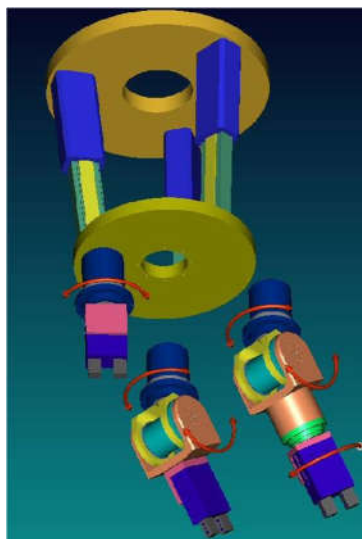


Figura 2.11 – Conceito de formação de robôs modulares [Larizza et al., 2006].

Ademais, o autor esclarece que um sistema modular apresenta características específicas, como, por exemplo, a escalabilidade e a comunicação determinística entre os diferentes módulos que compõem o robô industrial. No trabalho, para descrever o princípio de funcionamento de robôs modulares, é descrito que, ao montar um conjunto de três de unidades básicas (um do tipo atuação prismática e dois do tipo atuação rotacional), é possível formar uma grande variedade de estruturas cinemáticas seriais ou paralelos. Para exemplificar, a Figura 2.11 ilustra um robô de estrutura combinada serial e paralela.

Com a finalidade de descrever o funcionamento de um módulo mecatrônico, os quais são aplicados a robôs modulares, são abordados conceitos, como, os aspectos construtivos mecânicos, de controle eletrônico e de comunicação. A energia utilizada no acionamento de cada módulo mecatrônico, que é do tipo elétrica. Com relação às unidades modulares, explica-se que as mesmas são agrupadas através de módulos passivos, obtendo, portanto, a estrutura desejada. A Figura 2.12 ilustra a formação de diferentes estruturas cinemáticas por meio de módulos ativos e passivos acoplados [Larizza et al., 2006].

No entanto, mesmo que esse trabalho apresente um robô modular com estrutura flexível e escalável, o sistema proposto não considera os conceitos de agentes no sistema de gerenciamento de sua arquitetura. Nessa pesquisa, o autor utiliza um software de código aberto denominado OROCOS (*Open RObot COntrol Software*), que realiza o gerenciamento dos módulos, a interface com os usuários, o controle local de cada módulo e a sincronização das

entidades através da rede de alta velocidade (determinista). O OROCOS, por meio da organização de suas tarefas, define o módulo que irá executar o planejamento de trajetória, definido previamente, e interagir com o usuário. Tais características são determinadas no momento da montagem, através dos pinos de encaixe.

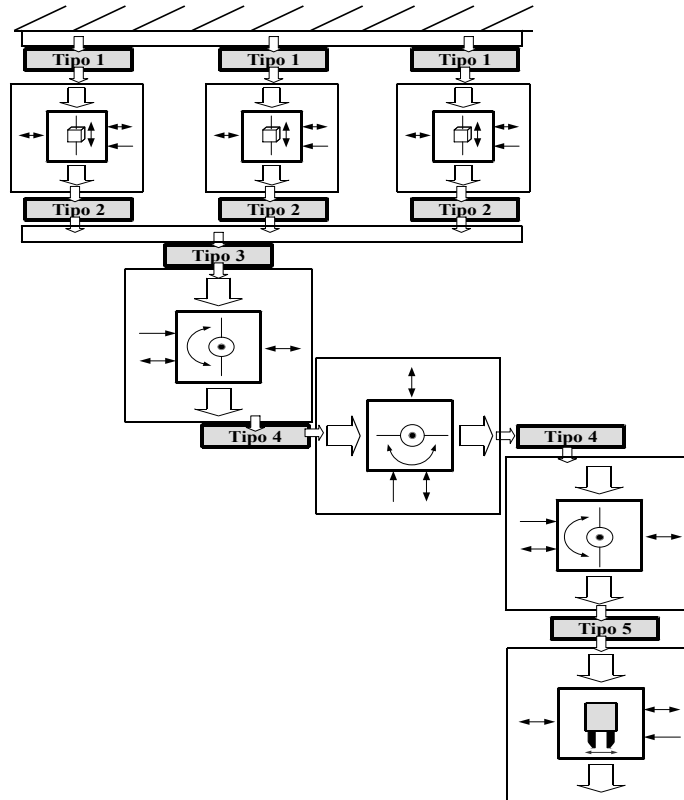


Figura 2.12 – Configuração modular cinemática paralelo-serial.

A Figura 2.13 apresenta o princípio de funcionamento do software proposto em função do hardware do robô modular [Larizza et al., 2006].

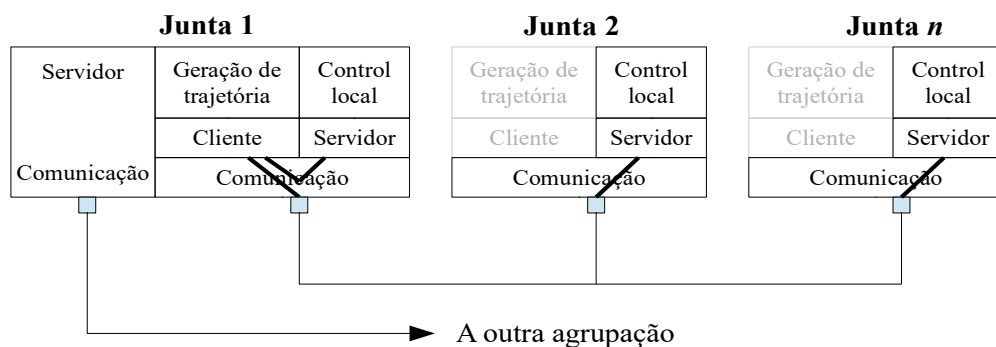


Figura 2.13 – Organização do software em função do hardware do robô modular.

Em síntese, o trabalho visa atender as necessidades associadas às rápidas mudanças nas linhas de produção de eletrodomésticos, as quais exigem cada vez mais a utilização de novos tipos de máquinas. Propõe-se a utilização de um robô modular formado por um número limitados de peças, com o objetivo de realizar a maioria das tarefas requeridas nessas linhas de produção (manipulação, solda, *pick-and-place*, etc.). O funcionamento deste sistema baseia-se em módulos mecatrônicos de uso industrial, são compostos por componentes mecânicos, elétricos, computacionais e de comunicação. É importante destacar que, na arquitetura de controle destes tipos de manipuladores, é utilizada a metodologia de controle distribuído.

Cavalcante, 2012, aborda o conceito de linha de produção como em um sistema composto por entidades ou agentes mecatrônicos, que são estações de trabalho governadas por um sistema Multiagente. Propõe-se a utilização de módulos mecatrônicos em linhas de manufatura, onde cada estação de trabalho é uma entidade autônoma (por exemplo uma fresadora, uma estação de solda, etc.) e estas estações estão baseadas na utilização de agentes como sistema de gerenciamento. Nesse trabalho, que é parte do projeto IDEAS da União Europeia [Maffei e Onori, 2011], o autor faz uma abordagem de demonstração e implementação dos sistemas Multiagentes dentro de uma plataforma virtual, obtendo, assim, uma linha de produção “reativa”, capaz de ser facilmente reconfigurada, ante a necessidade de fabricação de novos e variados produtos. Ainda, o sistema proposto apresenta características tais como: reatividade, modularidade, escalabilidade, adaptabilidade, evolucionabilidade (aprendizagem), entre outros. Como resultado, é apresentado uma linha de produção, na qual apresenta baixos tempos de reconfiguração, sem perdas de rendimento.

Mesmo que não tenham sido abordados assuntos relacionados à robótica, nesta pesquisa, foram mencionados conceitos, tais como: módulos mecatrônicos, sistemas modulares, agentes mecatrônicos e Sistemas Multiagentes, destacando-se uma proposta de trabalho inteiramente estruturado em controle distribuído.

Em Peixoto, 2012, que trata-se de uma análise similar a Cavalcante, 2012, utiliza-se uma arquitetura orientada a serviços e baseadas em agentes, as quais são aplicadas na organização e otimização de uma linha de produção. A orientação a serviços consiste na descentralização do controle e a disposição de componentes de forma a se agruparem virtualmente e formar novos dispositivos, com novas funcionalidades, otimizando investimento, recursos e tempo de produção. Nesta abordagem, as estações da linha de produção comportam-se como agentes de manufatura, de modo que, em um universo de Multiagentes, negociam entre si as tarefas a serem realizadas. Este procedimento, entre outras finalidades, tem como objetivo atender às adaptações na linha devido à variação de um determinado

produto, o que reduz o tempo de troca de processos. Embora, do ponto de vista de implementação, a análise seja interessante, neste trabalho não é discutido o uso de módulos mecatrônicos, bem como os assuntos relacionados à robótica. Além disso, é importante destacar que essa abordagem se baseia em uma arquitetura de controle distribuído, aplicada a sistemas industriais.

Ferreira et al., 2012, definem conceitos e métodos de organização de agentes, os quais, seguidamente, são aplicados a módulos mecatrônicos de uso industrial. Esta pesquisa, assim como Cavalcante, 2012, faz parte de um projeto da União Europeia. Em relação à interação dos agentes, os autores analisam os seguintes aspectos: a formação das coalisões, a troca de informações, suas habilidades e, por fim, o controle e as adaptações no processo produtivo. Desta forma, o trabalho apresenta uma estratégia de visualização dos diferentes módulos mecatrônicos dentro de uma linha de produção, o que representa uma equivalência com um supervisor de um sistema de controle centralizado convencional. Não são abordados os assuntos relacionados à robótica ou módulos mecatrônicos aplicados nessa área, isto é, este trabalho foca no uso de agentes mecatrônicos em ambientes industriais e na supervisão em suas atividades.

Ulewicz et al., 2012 apresentam o projeto e, a implementação e a avaliação de uma abordagem híbrida de agentes aplicados em automação de sistemas de fabricação. Esta proposta baseia-se, principalmente, na organização funcional da estrutura de software das unidades mecatrônicas em duas camadas hierárquicas (agentes funcionais de alto nível e agentes de baixo nível), conforme os padrões da FIPA, tal como apresentado na Figura 2.14 [Ulewicz et al., 2012].

Os agentes de alto nível consistem nos agentes de sistema, que são encarregados na formação de outros agentes, na definição das tarefas a serem realizadas e na execução das atividades administrativas, e nos agentes de hardware, os quais realizam as atividades de controle de hardware dentro da linha de produção e executam funções que requerem processamento em tempo real.

A plataforma de baixo nível de agentes, em grande parte de suas aplicações, é utilizada na interação e percepção do ambiente por meio de dispositivos atuadores e sensores. Neste nível, o autor propõe a utilização de três tipos de agentes: os agentes de sistema, os agentes de hardware e os agentes de comunicação [Ulewicz et al., 2012].

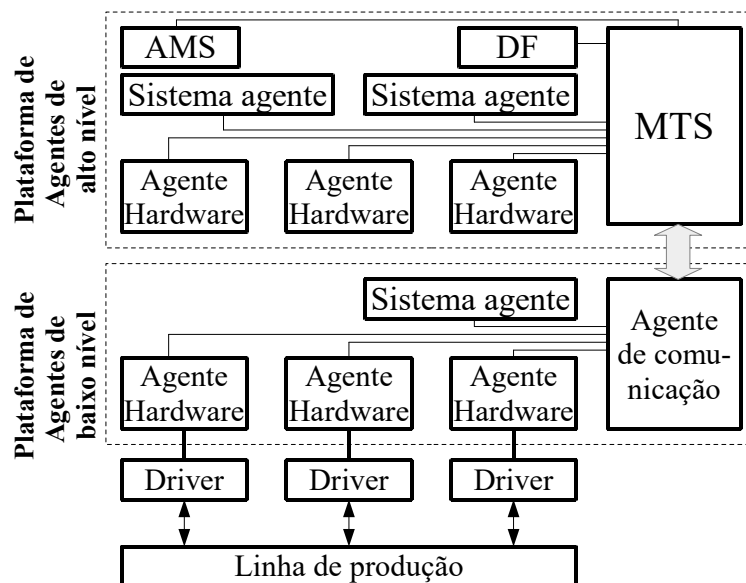


Figura 2.14 – Arquitetura de software de um sistema Multiagentes.

Os agentes de sistema possuem capacidades similares aos agentes de sistema de alto nível, entre elas, a organização dos agentes de baixo nível, a execução de funções específicas de gerenciamento de atividades, entre outras. Os agentes de comunicação são responsáveis pelo envio de mensagens entre os níveis de agentes (alto e baixo nível) e pela capacidade de processar estes dados. Os agentes de hardware de baixo nível, diferentemente dos agentes de alto nível, são entidades destinadas a uma tarefa específica, como, por exemplo, os atuadores. Além disso, como pode-se observar na Figura 2.14, esse trabalho representa a organização dos agentes de acordo com o padrão da FIPA. Nesta norma, existem módulos complementares que, embora não sejam tratados neste trabalho, são considerados pelo autor, tais como, o MTS (sistema de transporte de mensagem), o DF (diretório facilitador de habilidades dos agentes) e o AMS (sistema de gerenciamento de agentes).

Esse trabalho trata, portanto, de uma abordagem de gerenciamento de agentes aplicados a uma linha de produção, de modo que o autor não menciona alguns conceitos importantes para este trabalho, tais como, as aplicações de agentes na área da robótica e as montagens de robôs a partir de módulos mecatrônicos. Também, este sistema é constituído por uma arquitetura de controle distribuído, a qual é aplicada em uma linha de produção industrial. As estações de trabalho são definidas como módulos mecatrônicos, que, neste trabalho, não foram especificadas suas características.

Com relação à proposta de uma plataforma de um robô modular baseado em agentes, que é o tema principal deste trabalho, é possível observar que, neste capítulo, existem autores que não tratam o assunto referente à montagem de robôs modulares, entre eles, Cavalcante,

2012; Ferreira et al., 2012; Ulewicz et al., 2012. Porém, mesmo que estes autores não trataram especificamente deste tema, é importante mencionar que tais contribuições são relevantes para o desenvolvimento desta pesquisa, pois estudam e implementam os agentes em sistemas industriais. Ainda que apresentados por diferentes estratégias na implementação de agentes em sistemas de manufatura, estes trabalhos procuraram obter linhas de produção com características como a capacidade de adaptação, a escalabilidade, a interação entre as entidades, a rápida reconfiguração, entre outras. Estas características são interessantes para a presente pesquisa, pois, no caso de robôs modulares, muitas aplicações realizadas nas linhas de manufatura apresentam similaridades.

Quanto aos trabalhos que tratam especificamente de robôs manipuladores de origem modular, pode-se mencionar que existem poucos autores que desenvolveram pesquisas nesta área. Entre os tipos de abordagem, pode-se mencionar: a construção de módulos mecatrônicos, a arquitetura de controle distribuída, o sistema eletrônico de processamento e a comunicação de alta eficiência aplicados a robôs modulares. No entanto, apesar desses trabalhos abordarem conceitos importantes para o desenvolvimento, observou-se que não foram utilizados agentes como sistema informático de gerenciamento dos módulos, conforme será apresentado a seguir.

O trabalho do Chen e Yang, 1996, propôs a montagem de robôs manipuladores de origem modular para pequenas aplicações. Em relação à montagem mecânica desse sistema, é importante destacar que existem similaridades com o assunto proposto neste trabalho. Dentre as características consideradas relevantes pode-se citar a construção dos módulos mecatrônicos e o método de união para a formação de um robô industrial. Atta-Konadu et al., 2005, aborda a implementação do controle distribuído para um robô de uso industrial, apresentando a capacidade de escalabilidade deste sistema. Neste caso, cada elo do robô é composto por um sistema eletrônico e um de atuação, separando cada elo do robô como uma unidade controladora independente. Desta forma, aqui é apresentado diversas similaridades com a presente pesquisa, como, por exemplo, o controle descentralizado, a escalabilidade e a definição de diferentes tipos de algoritmos controladores para cada módulo.

Finalmente, um dos trabalhos que mais motivaram esta pesquisa foi realizado por Larizza et al., 2006. O trabalho, utiliza módulos mecatrônicos na formação de robôs manipuladores de uso industrial, apresentando características, como a alta capacidade de adaptação, a escalabilidade, a capacidade de formação de diferentes tipos de estrutura cinemáticas e a capacidade de processamento de dados dentro dos seus módulos. Embora o trabalho trate de conceitos importantes na área de robô modulares de uso industrial, o autor não

utiliza o conceito de agentes como base do sistema de gerenciamento dos módulos mecatrônicos.

3 MÓDULOS MECATRÔNICOS BASEADOS EM AGENTES

Este capítulo descreve as características dos módulos mecatrônicos auto organizáveis, baseados nos conceitos de agentes que são necessários na formação de robôs manipuladores de cadeias cinemáticas seriais. Nesta pesquisa, o estudo das unidades modulares abrange aspectos relacionados ao projeto mecatrônico, à descrição das funcionalidades dos módulos mecatrônicos do ponto de vista da teoria de agentes, além da proposta para o controle e planejamento de trajetória da estrutura formada.

Como já afirmado, os módulos mecatrônicos são fundamentais na formação de sistemas mecânicos de maior complexidade, como, por exemplo, robôs manipuladores. Desta forma, para que tais sistemas sejam eficazes na montagem de manipuladores seriais, eles devem apresentar determinadas características, tais como as seguintes:

Homogeneidade: Cada módulo deve ser composto por um hardware eletrônico e por um software dedicados ao seu gerenciamento;

Controle local: O comportamento de uma unidade mecatrônica deve ser determinada pelas informações locais do sistema e de outras unidades, permitindo, assim, operações em forma distribuída;

Reconfiguração dinâmica: as unidades devem ser capazes de se ajustar automaticamente à estrutura robótica formada, dependendo de sua localização dentro da cadeia cinemática.

Em síntese, para caracterizar um módulo mecatrônico na construção de robôs modulares, o presente trabalho aborda os seguintes aspectos:

1. Definição e descrição dos requisitos de projeto mecânicos para o desenvolvimento de um módulo mecatrônico;
2. Proposta para o reconhecimento da estrutura cinemática formada pelos módulos mecatrônicos;
3. Proposta para o controle e planejamento de trajetória do sistema robótico modular;
4. Proposição de uma arquitetura Multiagentes para o gerenciamento dos módulos mecatrônicos.

3.1 Metodologia de projeto dos módulos mecatrônicos

Nesta pesquisa, para descrever os módulos mecatrônicos na construção de robôs modulares, utilizam-se as metodologia de projeto proposta por Pahl et al., 2007; Rios, 2009; conforme descrito a seguir.

1. **Especificação de Projeto:** Consiste nas informações coletadas pertinentes ao produto a ser desenvolvido, definindo por meio de requisitos de projeto e das metas funcionais a serem atingidas pelo produto.
2. **Projeto Conceitual:** Trata da consolidação dos conceitos tecnológicos funcionais e morfológicos dos módulos mecatrônicos.
3. **Projeto Preliminar:** Especifica as informações técnicas referentes aos módulos mecatrônicos.
4. **Projeto Detalhado:** Baseia-se na documentação para a produção industrial e a avaliação dos módulos mecatrônicos. Por se tratar de um trabalho que não apresenta uma abordagem experimental, esta etapa não será discutida no desenvolvimento desta seção.

3.1.1 Especificações do projeto

Para definir os requisitos de projeto de um modulo mecatrônico elementar, primeiramente, é necessário estabelecer as características de um robô modular com estrutura cinemática serial.

Os robôs modulares de uso industrial, quanto ao seu princípio de funcionamento, são constituídos por módulos ativos (módulos mecatrônicos) e módulos passivos (*links* mecânicos). Ademais, estes robôs caracterizam-se por necessitarem de poucos componentes para o seu funcionamento, ou seja, para configurar este sistema, além dos módulos ativos e passivos, são necessários elementos mecânicos de fixação, uma rede local Ethernet e uma fonte de energia para o seu acionamento. Os módulos mecatrônicos, na maioria de suas aplicações, são acionados por motores elétricos, que realizam movimentos lineares ou rotacionais. Atuadores com fontes de energias hidráulicas e pneumáticas são pouco utilizados no acionamento de robôs industriais, porém, em certas aplicações, apresentam características desejáveis, como, por exemplo, a propriedade de resiliência em atuadores pneumáticos nas tarefas de manipulações de peças e possibilidade de operar em ambientes classificados com risco de explosão.

Para o funcionamento de um módulo mecatrônico elementar, no contexto do presente trabalho, são consideradas as seguintes especificações de projeto:

1. Cada módulo mecatrônico constitui um grau de liberdade (junta) para o movimento do robô;
2. Um módulo mecatrônico deve dispor de um espaço adequado para posicionar um sistema de controle digital;
3. Devem ser capazes de efetuar acionamentos lineares;
4. A ligação entre os módulos ativos é feita através de um módulo passivo, o qual realiza a fixação mecânica entre os dispositivos. Nesta abordagem, este dispositivo também é denominado como *módulo auxiliar*.
5. Os módulos auxiliares são responsáveis pela ligação entre os módulos mecatrônicos e o chão de fábrica e os efetuidores.
6. Os sensores e válvulas direcionais utilizados nos atuadores pneumáticos deverão ser esquematicamente similares para atuadores lineares e rotacionais.
7. A resolução de leitura de posicionamento dos eixos deverá ser apropriada para movimentos precisos (por exemplo, uma resolução superior aos 0,1 mm ou 0,09 graus sexagesimais).
8. A velocidade de movimentação linear compatível com robôs de acionamento pneumático (por exemplo de 1 m/s e a velocidade rotacional máxima é de 1,57 rad/s).
9. A ligação mecânica pneumática é realizada através dos módulos auxiliares.

3.1.2 Projeto Conceitual

Esta seção objetiva determinar a concepção conceitual, tecnológica, funcional e morfológica de um módulo mecatrônico, através da seleção, criação e representação da solução do problema proposto.

Os sistemas modulares apresentam propriedades de flexibilidade e capacidade de reutilização para diferentes tarefas. No entanto, tais sistemas exigem alto grau de padronização, tanto em interfaces mecânicas, quanto eletrônicas. Em um sistema modular, quando configurado como um robô manipulador, cada módulo deve ser capaz de mudar seu modo de operação para diferentes configurações cinemáticas, dependendo da sua localização dentro da estrutura. Os principais componentes eletromecânicos envolvidos nos módulos mecatrônicos propostos são descritos a seguir.

Estrutura mecânica

Os módulos são unidades que, dentro de uma mesma estrutura física, são compostos por um elemento atuador, uma válvula proporcional, sensores e componentes eletrônicos de controle. Os componentes eletrônicos e de atuação devem estar protegidos por um perfil metálico (resistência mecânica e blindagem elétrica), possibilitando, assim, que o atuador se movimente de forma segura. A estrutura de suporte do módulo deve ser compacta e resistente, dispondo de furos para as conexões mecânicas e elétricas que permitem a ligação com outros módulos e facilitam o intercâmbio de informação e energia. Para exemplificar, a Figura 3.2, apresentada na Seção 3.1.3, ilustra a estrutura mecânica de um módulo de acionamento linear.

Atuador

Como os robôs modulares seriais podem ser representados por diferentes configurações cinemáticas, optou-se por utilizar cilindros pneumáticos como atuadores, pois, além de apresentarem uma boa relação potência/peso, são elementos de atuação linear e rotacional. Visando a simplificar as análises, na presente abordagem, os módulos mecatrônicos serão detalhados do ponto de vista mecânico como elementos de atuação linear, não tratando das características mecânicas dos sistemas de atuação rotacional.

Sensores

Neste trabalho, os sensores são responsáveis por fornecer as informações do ambiente com o qual estão interagindo as unidades mecatrônicas. Para um sistema modular, as variáveis de interesse são: pressão das câmaras do cilindro, posição do elemento efetuator e orientação do módulo mecatrônico. Detalhes sobre estes sensores, são apresentados na Seção 3.1.3.

Arquitetura de controle de posição, planejamento de trajetória e gerenciamento dos módulos mecatrônicos

No projeto de um sistema modular, uma das características importantes a serem consideradas é a definição do tipo de arquitetura utilizada para o controle de posição das unidades de atuação (módulos mecatrônicos), planejamento de trajetória e gerenciamento de tarefas dos módulos mecatrônicos. Para tanto, propõe-se a utilização, além de um algoritmo

clássico de controle (Proporcional Integral Derivativo), de um algoritmo para o controle de posição não linear (apresentados no ANEXO A – ALGORITMOS DE CONTROLE). Além disso, é representado um algoritmo de planejamento de trajetória para realizar a movimentação da estrutura e, por fim, um algoritmo de gestão e de comunicação dos módulos. Tais arquiteturas serão discutidas nas seções 3.3 e 3.4.

Sistemas de potência

Em função das características de acionamento das unidades mecatrônicas propostas, o sistema de potência a ser empregado é baseado em ar comprimido, muito difundido em ambientes industriais, de baixo custo e não agressivo para o meio ambiente. As fontes de energia utilizadas no acionamento de sistemas moduladores serão abordadas na Seção 3.1.3.

Módulos auxiliares de fixação

Quanto à montagem de sistemas modulares, uma das principais características de projeto que devem ser consideradas é a forma de conexão entre os módulos. Assim, são utilizados módulos auxiliares para efetuar a montagem de um manipulador, os quais consistem em elementos de fácil fabricação, robustos e de longa durabilidade nas diversas reutilizações.

Os módulos auxiliares caracterizam-se por facilitar a passagem de energia pneumática e mecânica. Para construir robôs modulares com diferentes configurações cinemáticas, propõe-se três tipos de módulos auxiliares em função do tipo de conexão:

Módulo auxiliar tipo 1: conexão mecânica pneumática entre a base fixa e o primeiro módulo mecatrônico;

Módulo auxiliar tipo 2: serve para realizar a ligação mecânica entre os módulos ativos;

Módulo auxiliar tipo 3: este tipo de conexão é de menor complexidade, servindo para realizar a conexão entre o efetuador e o último elo ativo da estrutura.

A Figura 3.1 apresenta a aplicação dos módulos auxiliares em um manipulador do tipo cartesiano. Estes tipos de módulos auxiliares serão detalhados na Seção 3.1.3.

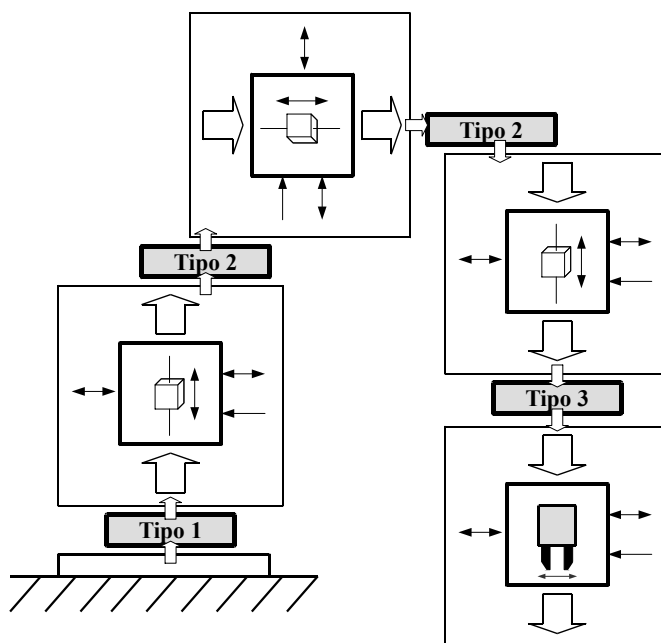


Figura 3.1 – Módulos auxiliares utilizados em um robô modular cartesiano

Efetuator robótico

Os efetadores são dispositivos fixados junto ao último elo e que permitem ao mesmo realizar uma determinada tarefa, assim como também são responsáveis pela operação final do robô. Normalmente estes elementos são projetados especialmente para uma tarefa ou aplicação, e a modo geral efetadores são denominados, como garras ou ferramentas. Estas garras podem ser do tipo mecânica (que utilizam mecanismos articulados para o acionamento por preensão) ou não mecânicas (garras de sucção ou ventosa, garras magnéticas, entre outras) [RIA, 2016]. Na abordagem deste trabalho não será tratado sobre este dispositivo.

Configuração dinâmica do sistema

Os módulos, quando agrupados, formam uma configuração geométrica com características dinâmicas que dependem das posições relativas de cada módulo mecatrônico. A configuração dinâmica da estrutura, a localização de cada unidade dentro desta estrutura e as orientações individuais, serão definidas mediante o procedimento descrito na Seção 3.2, que aborda um algoritmo de reconhecimento que tem como saída o volume de trabalho e a cinemática direta e inversa para uma determinada configuração mecânica.

3.1.3 Projeto Preliminar

Esta seção descreve as especificações e funcionalidades dos componentes mecânicos de um módulo mecatrônico.

O modelo proposto baseia-se em pesquisas desenvolvidas, as quais foram discutidas no Capítulo 2 deste trabalho. Quanto à estrutura mecânica, o modelo proposto não consiste de um modelo definitivo, de modo que será necessário realizar futuramente testes experimentais para sua validação.

Descreve-se a seguir os elementos do projeto preliminar de um módulo mecatrônico, sendo especificados os aspetos considerados mais importantes:

Estrutura mecânica do módulo mecatrônico

Os módulos mecatrônicos, com relação aos aspetos construtivos, são definidos por suas dimensões, pelo tipo de atuação e por suas características dinâmicas e elétricas. Ademais, quanto a sua estrutura, as unidades são representadas por módulos ativos, os quais apresentam capacidade de processamento de dados e atuação, e por módulos auxiliares, que são elementos mecânicos complementares aos módulos mecatrônicos. A Figura 3.2, apresenta um robô cartesiano com 3 GDL representado por módulos mecatrônicos e módulos auxiliares.

Para caracterizar uma estrutura robótica modular, cada entidade mecatrônica deve atender a requisitos como: rigidez mecânica, capacidade de fornecimento de energia para os módulos, capacidade de processamento de dados e interação com os demais módulos.

Para tanto, cada módulo mecatrônico deve ser composto pelos seguintes componentes:

1. Cilindro de acionamento pneumático;
2. Circuito eletrônico de controle e comunicação;
3. Válvula de acionamento proporcional pneumático;
4. Sensores de pressão, posição e aceleração;
5. Conectores mecânicos e elétricos;

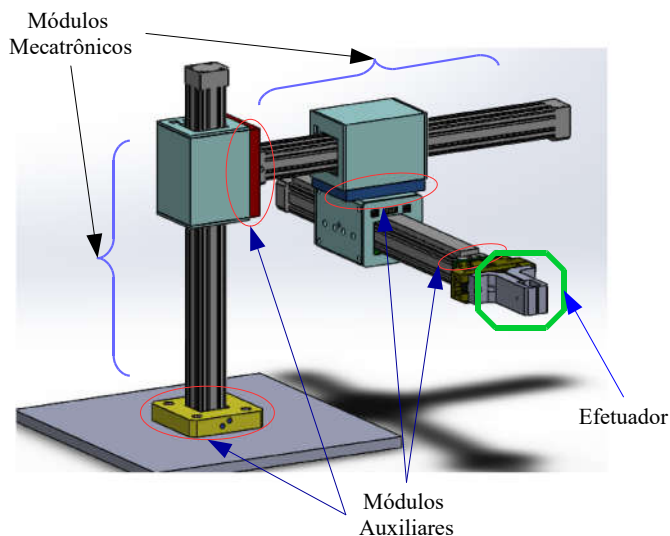


Figura 3.2 – Estrutura serial de um robô cartesiano modular

Um módulo mecatrônico, quanto ao seu princípio de funcionamento, consiste de um sistema com capacidade de tomada de decisões para o seu acionamento. Os movimentos do atuador linear são provenientes da ação de uma válvula proporcional pneumática, enquanto que o sinal de controle é obtido por meio de um circuito eletrônico que processa diferentes dados do módulo mecatrônico e do seu entorno. Desse modo, como os módulos mecatrônicos tratam-se de sistemas compostos por elementos acionados por diferentes fontes de energia (elétrica, mecânica e pneumática), é necessário estabelecer o posicionamento adequado dos componentes a fim de atender os limites físicos da estrutura e, conseqüentemente, as características operacionais dos componentes a serem utilizados no sistema. Os critérios considerados na presente pesquisa são:

- a) Os componentes eletromecânicos devem ser alojados no interior do módulo;
- b) O atuador de deslocamento corresponde a um cilindro pneumático;
- c) As conexões de energia pneumáticas são realizadas através dos módulos auxiliares;
- d) As conexões elétricas e de dados, são realizadas utilizando conectores externos do tipo RJ-45 (dados) e DB9 (energia);
- e) Os módulos auxiliares dispõem de pinos físicos que ativam um contato elétrico no módulo mecatrônico, determinando, assim, a posição relativa na configuração cinemática.

A Figura 3.2 ilustra um modelo de estrutura com módulo mecatrônico de atuação linear, e na Figura 3.3 um esquema detalhado do módulo e seus componentes internos.

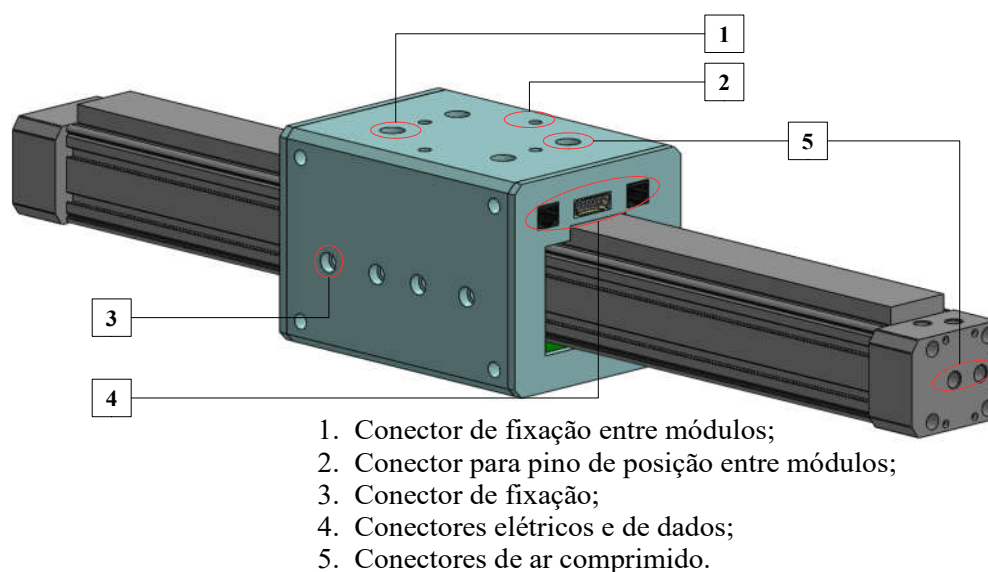


Figura 3.3 – Módulo mecatrônico de acionamento linear

Atuadores pneumáticos

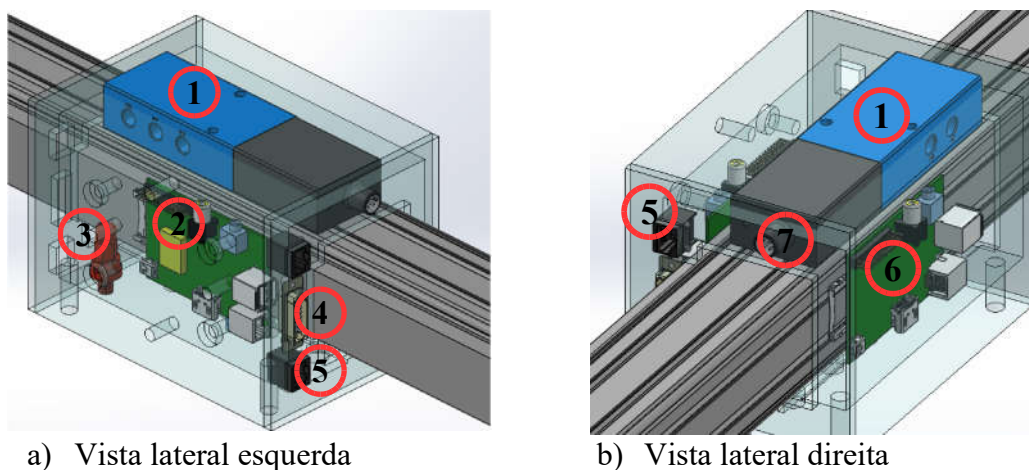
A fim de atender os requisitos de funcionamento de um módulo mecatrônico, estabelece-se que os atuadores devem ser de dupla ação e sem haste, pois facilitam a montagem dos componentes eletrônicos, bem como a interligação entre os demais módulos mecatrônicos do sistema. Além disso, este componente caracteriza-se por boa relação potência/peso, além de dispensar o uso de sistemas de transmissão mecânica. Para exemplificar, apresenta-se o modelo DGPIL-32-300-PPV-AIF-KF-GK da Festo®, que possui interface de comunicação para leitura de sua posição absoluta. Na Tabela 3.1 são apresentadas as características deste atuador.

Como ilustrado na Figura 3.4, o carro central de cargas dispõe de uma superfície apropriada para a instalação dos componentes eletrônicos e de comando, assim como um espaço para realizar as conexões de fixação entre os módulos mecatrônicos e auxiliares. Cabe destacar que futuras pesquisas, em relação ao servoposicionador pneumático, estão desenvolvendo sensores de pressão que serão integrados no embolo do próprio cilindro, permitindo maior compactação e minimização da utilização do espaço interno.



Figura 3.4 – Módulo mecatrônico e os componentes internos

Uma disposição interna dos componentes de um módulo mecatrônico é apresentada na Figura 3.5. Os dispositivos encontram-se fixados no suporte metálico e no carro de carga, nesta figura não são representados aos cabos de alimentação e dados de cada elemento. A estrutura metálica se apresenta como transparente para apreciar a visualização dos principais componentes.



1) Servoválvula; 2) Raspberry Pi Modelo B; 3) Sensor de pressão; 4) Conector de alimentação elétrica; 5) Conector RJ45; 6) Condicionamento de sinais; 7) Conector servoválvula

Figura 3.5 – Alocação dos componentes de um módulo mecatrônico

Tabela 3.1 – Especificações técnicas dos atuadores DGPIIL [Festo, 2016]

| Características | Valores |
|--|-----------------------|
| Pressão Trabalho | 4 – 8 bar |
| Força Teórica a 6 bar Linearidade | 483 N |
| Conexão Pneumática | G1/8 |
| Comprimento Amortecimento | 0,020 m |
| Máxima velocidade permitida | 2 m/s |
| Tensão de alimentação do sensor de posição | 24 ($\pm 25\%$) Vcc |
| Resolução | $\leq 0,01$ mm |
| Interface de Medição da Posição do êmbolo | Digital, CAN |
| Consumo de Corrente | 90 mA |

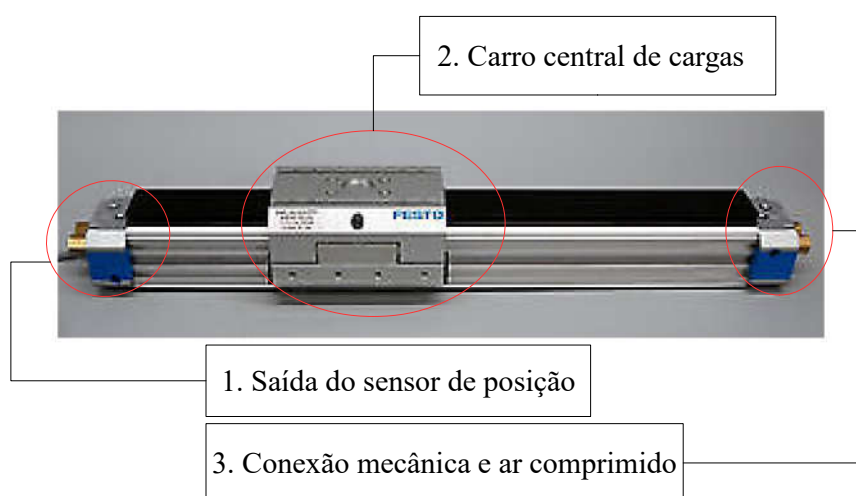


Figura 3.6 – Cilindro sem haste de atuação pneumático Festo

Circuito de controle e comunicação

O circuito eletrônico, em módulo mecatrônico, é encarregado pelo processamento de dados, pela tomada de decisões e pela interface entre a entidade e o mundo exterior. Tais circuitos, basicamente, são compostos por dois principais componentes, no caso, o Raspberry Pi Modelo B e o Microcontrolador PIC18F4580, os quais foram explicados no capítulo anterior. As características técnicas destes componentes estão detalhadas na Tabela 3.2 e na Tabela 3.3.

Tabela 3.2 – Características do Raspberry Pi Modelo B

| | |
|---------------------|---|
| Chip SOC | Broadcom BCM2835 full HD processador de aplicações multimídia |
| CPU | 700 MHz ARM1176JZF-S core (ARM11 family) |
| GPU | Dual Core VideoCore IV® Multimedia Co-Processor, 1080p30 decodificador h.264/MPEG-4 |
| Memoria (SDRAM) | 512MB RAM (compartilhada com GPU) |
| Portas USB | 2 portas USB |
| Saídas de vídeo | HDMI (rev 1.3 & 1.4) Composite RCA |
| Rede | 1 porta ethernet |
| Potência de consumo | 700mA (3.5W) |
| Alimentação | 5V MicroUsb |
| Dimensões | 85.60 mm × 53.98 mm |
| Sistema operativo | Debian GNU/Linux, Fedora, Arch Linux, Raspbian19 |
| Módulos | 2 x SPI, I2C, UART, PWM |

Tabela 3.3 – Características do microcontrolador PIC18F4580

| | |
|-------------------------------|--------------|
| Memória de programa | 32 Kbytes |
| Memória RAM | 1,536 Kbytes |
| EEPROM | 226 bytes |
| Porta I/O | 36 |
| Temporizador de 16 bit | 3 |
| ADC interno 10 bits (100 kps) | 11 canais |
| Tensão de Trabalho | 2 – 5.5Vcc |
| USART | Compatível |
| SPI | Compatível |
| ECAN | Compatível |

Sensores de pressão e aceleração

Construtivamente, os módulos mecatrônicos são constituídos por dois sensores de pressão absoluta e um sensor de aceleração. O presente trabalho propõe o uso dos sensores MPXV5050GP, para medição de pressão absoluta, e ADXL316WBCSZ, para medição de aceleração, pois, além de suas dimensões compactas, apresentam saídas digitais com 12 bits de resolução e uma boa relação sinal / ruído.

Sistemas de potência

O ar comprimido consiste de um recurso muito difundido em ambientes industriais, caracterizado pela facilidade de instalação e distribuição e pela baixa agressividade ao meio ambiente. Quando utilizado em aplicações industriais, o ar comprimido deve ser filtrado para eliminar suas impurezas e umidades, de modo que a pressão de suprimento deve estar entre 6 e 8 Pa.

Como a energia elétrica será utilizada na alimentação das unidades de controle de baixa potência, é necessária uma fonte externa de 24V e 2A para a alimentação de uma estrutura com até 3 módulos.

3.1.4 Acoplamento entre módulos mecatrônicos

Há, na construção de sistemas modulares seriais, três formas de acoplamento entre os módulos mecatrônicos mediante a utilização de dispositivos mecânicos, que, neste trabalho, são denominados como módulos auxiliares: a) Acoplamento entre módulos mecatrônicos (apresentado na Figura 3.7); b) Acoplamento entre um módulo mecatrônico e elemento terminal (efetuador), c) Acoplamento entre a Base fixa e um módulo mecatrônico.

Quanto aos acoplamentos, é importante ressaltar que, os módulos mecatrônicos, são fixados por meio de porcas e parafusos mediante o uso de módulos auxiliares, permitindo a troca de esforço mecânico e energia pneumática.

A energia elétrica e os elementos de transmissão de dados devem ser conectadas através de conectores específicos, separados do módulo auxiliar. Isto deve-se principalmente, de se tratar um projeto preliminar, e ainda estes desafios deverão ser encarados em inovações futuras do equipamento.

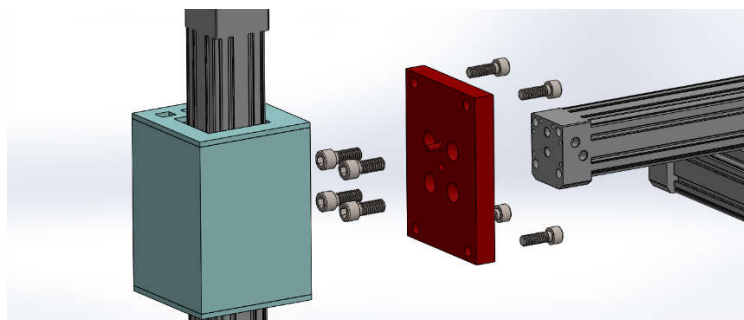


Figura 3.7 – ligação entre dois módulos mecatrônicos

Tipos ou módulos auxiliares

Como já mencionado, os robôs modulares são constituídos de módulos ativos (ou módulos mecatrônicos) e módulos passivos (módulos auxiliares). Estes últimos, basicamente, permitem a ligação física entre os módulos mecatrônicos, formando, neste caso, as configurações cinemáticas convencionais de manipuladores industriais. Desta forma, é importante destacar que deve haver um tipo de módulo auxiliar específico para cada tipo de acoplamento entre os módulos mecatrônicos.

No sentido de conectar diferentes módulos mecatrônicos, dependendo de sua aplicação, propõe-se a utilização de três tipos módulos auxiliares:

Módulo auxiliar tipo 1: realiza a conexão mecânico pneumática entre a estrutura de suporte mecânico ao sistema (base) e o primeiro módulo mecatrônico (1 GDL), tal como apresentado na Figura 3.8. Os quatro furos laterais são destinados ao acoplamento do módulo auxiliar e a base, já o módulo mecatrônico é fixado mediante um parafuso na parte inferior. Os furos laterais são específicos para a conexão do ar comprimido. Cabe destacar que este módulo auxiliar está definido para uma junta do tipo prismática.

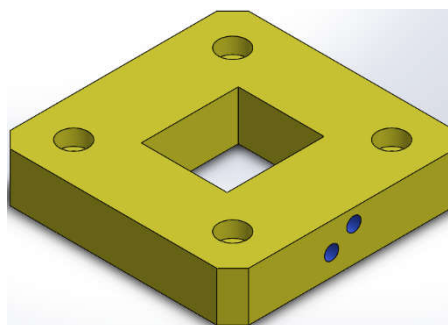


Figura 3.8 – Módulo auxiliar tipo 1

Módulo auxiliar tipo 2: permite a ligação mecânica entre os módulos prismáticos de forma precisa, possibilitando o movimento relativo entre dois módulos mecatrônicos, conforme ilustrado na Figura 3.9.

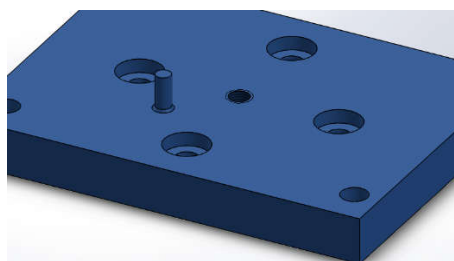


Figura 3.9 – Módulo auxiliar tipo 2

Da mesma forma que no caso anterior, existem quatro furos para a fixação de cada módulo auxiliar, e no centro do módulo auxiliar é apresentado o conector relativo a transferência de energia pneumática. Como pode ser observado, o pino metálico apresentado, é utilizado para identificar fisicamente a orientação da conexão relativa entre os módulos mecatrônicos. Isto é realizado mediante um chaveamento elétrico no sistema modular.

Módulo auxiliar tipo 3: estabelece a conexão entre o efetuator e o último elo ativo da estrutura, conforme apresentado na Figura 3.10.

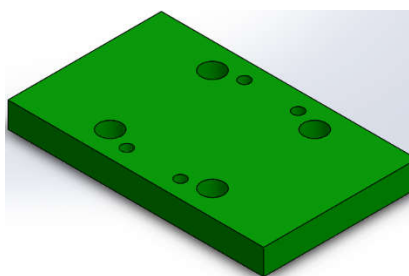


Figura 3.10 – Módulo auxiliar tipo 3

3.2 Definição da cinemática da estrutura formada pelos módulos mecatrônicos

Para Barrientos, 1997, a cinemática de um manipulador consiste na descrição analítica do movimento espacial do robô em função do tempo, considerando as relações entre a posição e orientação do efetuator do robô e os valores de suas respectivas coordenadas articulares. No que diz respeito à cinemática de manipuladores, enfoca-se apenas nos manipuladores seriais, visto que estas consistem nas configurações mais utilizadas em ambientes industriais. É importante destacar que, embora os módulos mecatrônicos possam formar diferentes tipos de

geometrias articulares (série, paralelo e série-paralelo), este trabalho apresenta uma abordagem de montagem e gerenciamento de sistemas com arquiteturas seriais.

Quanto aos estudos de cinemática, tratando-se de robôs manipuladores ou robôs modulares, existem, basicamente, dois problemas a serem resolvidos: a cinemática direta e a cinemática inversa. A primeira consiste na determinação da posição e orientação do extremo do manipulador relacionado ao sistema de referência da estrutura. Para tanto, é necessário conhecer os valores das articulações e os parâmetros geométricos do robô. Para resolver os problemas associados à cinemática direta, autores como Fu et al., 1987; Siciliano et al., 2010, entre outros, sugerem o uso do método de Denavit-Hartenberg, o qual é utilizado para descrever e apresentar as relações cinemáticas entre um elo e seus elos adjacentes, de forma que são necessários quatro parâmetros por elo para a definição completa dos mesmos. A cinemática inversa baseia-se em definir a configuração das articulações de um robô para uma determinada posição e orientação do efetuador conhecidas [Barrientos, 1997].

A montagem de sistemas modulares, levando em conta tanto o volume de trabalho de cada arquitetura quanto a ordem das posições de cada módulo mecatrônico na cadeia serial, deve possibilitar a determinação das matrizes de cinemática da estrutura em questão. Para tanto, é necessária a identificação automatizada dos módulos e de sua respectiva posição dentro do sistema mecânico, essa propriedade, além de facilitar a coordenação e supervisão do sistema (agente que representa ao robô) no seu conjunto, facilita o estabelecimento das diretrizes das tarefas, como, por exemplo, o planejamento da trajetória, necessário para o controle global de movimentação do sistema.

3.2.1 Cinemática Direta

Nos sistemas modulares, como os cálculos das configurações cinemáticas são refeitos toda vez que o sistema é inicializado, a cinemática direta consiste na determinação da posição do efetuador a partir das diferentes possibilidades de configurações dos elos.

Em geral, um robô com n graus de liberdade é formado por n elos unidos por n juntas, de forma que cada par articulação + elo constitui um grau de liberdade do sistema. Cada elo do sistema está relacionado a um sistema de referência que, por meio das matrizes de transformações homogêneas, possibilita representar as rotações e translações relativas entre os distintos elos (módulos mecatrônicos) que compõem o manipulador, tal como apresentadas nas equações 3.1 e 3.2, onde ${}^{i-1}A_i$ é a matriz homogêneas associadas a cada módulo mecatrônico. Desta forma, a matriz 0A_1 descreve a posição e orientação do sistema de referência do primeiro

elo ou módulo, que é o sistema de referência da base. 1A_2 descreve a posição e orientação e posição do segundo elo respeito do primeiro, e assim por diante. Da mesma forma, denomina-se 0A_k as matrizes resultantes do produto das matrizes ${}^{i-1}A_i$ com i de 1 até k , é possível representar a cadeia cinemática do robô manipulador [Barrientos, 1997].

3.2.2 Determinação da matriz de transformação

O método da matriz de D-H possibilita representar de maneira sistemática as relações cinemáticas entre um elo e um conjunto de n elos adjacentes. No âmbito de sistema modulares, estas características são importantes pois é possível determinar por meio dos parâmetros de D-H a relação de n elos adjacentes em função do número de módulos mecatrônicos acoplados ao sistema.

A Tabela 3.4 apresenta os parâmetros de D-H para um sistema modular, de modo que, conforme varia o número de elementos (elos) acoplados ao sistema serial, os números dos parâmetros desta matriz também se alteram.

Como mencionado, os módulos mecatrônicos são entidades autônomas capazes de fornecer informações referentes a suas características para um sistema de gerenciamento, possibilitando, assim, o fornecimento destes dados para o sistema Multiagente quando for solicitado. Os dados, a nível de robô modular, são detalhados na Seção 3.4.

Tabela 3.4 – Parâmetros de Denavit-Hartenberg de um sistema modular.

| Elo | θ_i | d_i | a_i | α_i |
|-----|------------|-------|-------|------------|
| 1 | θ_1 | d_1 | a_1 | α_1 |
| 2 | θ_2 | d_2 | a_2 | α_2 |
| n | θ_n | d_n | a_n | α_n |

Desta forma, para determinar os parâmetros da matriz D-H, cada unidade deve disponibilizar as seguintes características:

- Comprimento do elo;
- Posição e orientação do módulo;
- Tipo de atuação;
- Tipo de conexão entre juntas.

Metodologicamente, a fim de correlacionar os parâmetros da matriz D-H com as características de unidade mecatrônica, propõe-se usar métodos diretos e indiretos na construção da matriz de D-H, conforme apresentado na Figura 3.11.

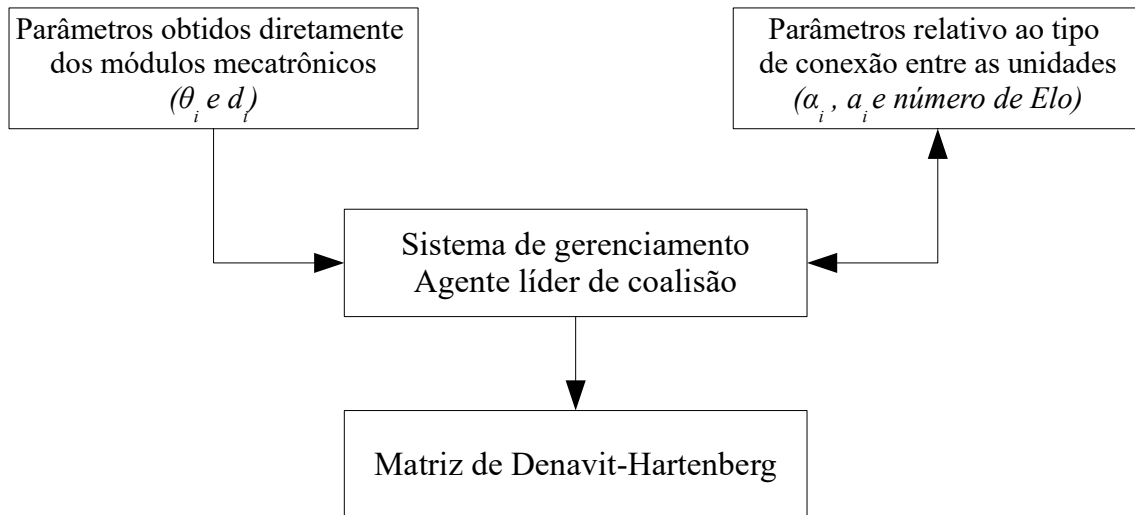


Figura 3.11 – Algoritmo de obtenção dos parâmetros de Denavit-Hartenberg

Método direto: Consiste na utilização dos parâmetros fornecidos por cada entidade, de modo que não há nenhum tipo de tratamento nestes dados. São utilizados pelo agente líder de coalisção para determinar a matriz de D-H e calcula-se as variáveis θ_i e d_i .

Método indireto: calcula-se os parâmetros α_i , a_i e o número do *elemento* associado a matriz D-H. Tais parâmetros são determinados pelo sistema de gerenciamento ou agente líder de coalisção por meio de dados fornecidos pelos agentes mecatrônicos. Neste caso, para determinar estes coeficientes, o agente utiliza as informações referentes às conexões entre os módulos, suas orientações relativas, ângulo de posicionamento e tipo de atuação.

3.3 Proposta para o planejamento de trajetórias controle para robôs modulares

Nesta seção, discute-se as propostas para que os robôs modulares possam executar as tarefas propostas. Um aspecto importante, além dos conceitos já mencionados, trata dos estudos de planejamento de trajetória e estratégia de controle, uma vez que estes tenham definida a sua estrutura geométrica e as suas características dinâmicas.

A seguir, são apresentadas as propostas para o planejamento de trajetórias e estratégias de controle aplicadas frequentemente em robôs manipuladores de uso industriais, as quais servem de base para o cálculo para os robôs modulares de cadeias seriais.

Nesta seção, discute-se as propostas para o planejamento de trajetórias e estratégias de controle a serem utilizadas em um robô modular constituído por cadeias cinemáticas seriais. Estas duas propostas são descritas separadamente a seguir.

3.3.1 Planejamento de trajetórias para robôs manipuladores

Em Biagiotti e Melchiorri, 2009, explica-se que o problema de planejamento da trajetória estabelece a relação entre dois elementos que pertencem a domínios diferentes, no caso, o tempo e o espaço. Assim, as trajetórias são, geralmente, expressas por meio de uma equação parametrizada em função do tempo, que fornece em cada instante a respectiva posição desejada. No entanto, após definir esta equação, existem outros aspectos relacionados com a sua aplicação que, apesar de não serem tratados neste trabalho, devem ser considerados, tais como discretização do tempo, saturação do sistema de acionamento, entre outros.

Na literatura, as trajetórias são abordadas de duas maneiras. A primeira delas refere-se ao planejamento do caminho, que indica o lugar geométrico dos pontos no espaço operacional, ou no espaço de junta, no qual o manipulador deve percorrer na execução do movimento lhe for atribuído. Já, o planejamento de trajetória consiste no caminho em que uma variável relacionada ao tempo é especificada em termos de velocidade ou acelerações em cada ponto [Siciliano et al., 2010].

Nesta perspectiva, levando-se em conta que o presente trabalho propõe apresentar uma abordagem relativa ao planejamento de trajetórias que atendam às funcionalidades dos sistemas modulares, a seguir, é apresentada a metodologia para a geração de trajetórias, as quais consideram as características de flexibilidade destes sistemas, conforme apresentado na Figura 3.12.

Inicialmente, após a montagem dos módulos mecatrônicos na configuração desejada para o robô, o sistema Multiagente é responsável por identificar as habilidades de cada unidade modular, definindo a configuração geométrica em questão (por exemplo, robô SCARA, cartesiano, entre outros). Definida a configuração, com o conhecimento dos algoritmos da cinemática direta e inversa, assim como os pontos intermediários (definidos previamente pelo usuário), estes pontos são interpolados mediante uma função *spline* de terceira ordem obtendo assim uma função de movimentação para cada junta. A trajetória gerada para cada junta, não leva em consideração suas restrições físicas ou o desvio de obstáculos no volume de trabalho.

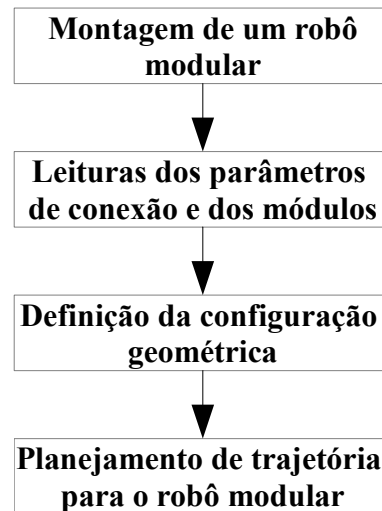


Figura 3.12 – Etapas para a abordagem do problema proposto

Splines de 3º grau (Cúbica)

Segundo Chapra e Canale, 2008, o objetivo das *splines* cúbicas consiste em obter um polinômio de terceiro grau para cada intervalo entre os nós, conforme a Equação 3.1.

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i. \quad (3.1)$$

Como, em $n+1$ dados, existem n intervalos, sendo cada intervalo definido com um polinômio da Equação 3.1. Para a obtenção das funções *spline* de terceira ordem (Figura 3.13), deve-se avaliar os coeficientes sendo necessárias $4n$ condições:

- a) os valores da função devem ser iguais nos nós internos ($2n - 2$ condições);
- b) a primeira e a última função devem passar através dos pontos extremos (2 condições);
- c) as derivadas de primeira ordem devem ser iguais nos nós internos ($n - 1$ condições);
- d) as derivadas de segunda ordem devem ser iguais nos nós internos ($n - 1$ condições);
- e) as derivadas de segunda ordem nos nós extremos são iguais a zero.

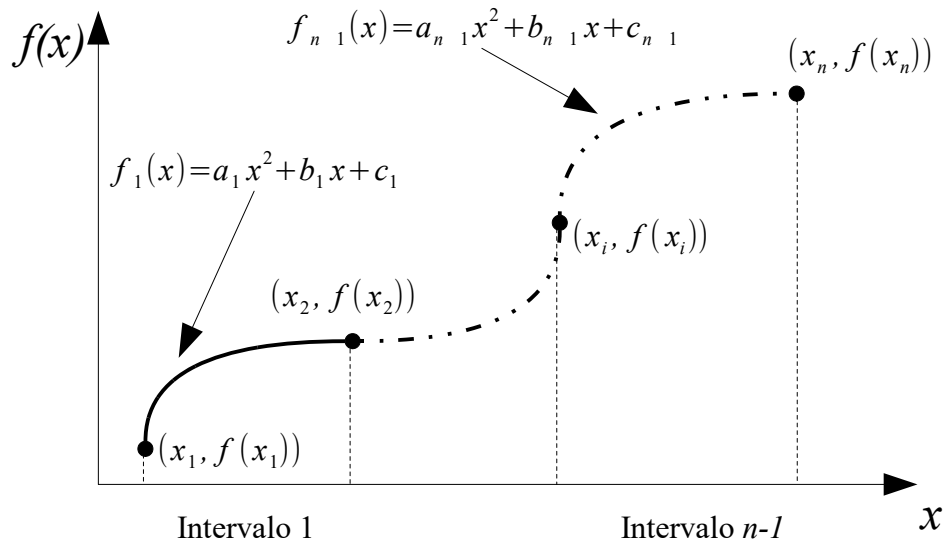


Figura 3.13 – Interpolação *spline* de 3º ordem

Chapra e Canale, 2008, indicam que a especificação dessas condições caracteriza as *splines* naturais. Se o valor da derivada de segunda ordem nos nós extremos não é zero, significa que existe alguma curvatura, ou seja, é possível utilizar esta informação de maneira alternativa para ter as duas condições finais.

Os autores também apresentam uma técnica alternativa, que, como apresentado no Capítulo 4, é muito utilizada no planejamento otimizado de trajetórias. Esta técnica requer a solução de apenas $n-1$ equações, e é representada da seguinte maneira:

Segundo Chapra e Canale, 2008, o primeiro passo para obtenção das *splines*, apresentadas por Cheney e Kincaid, 2012, é considerar que cada par de pontos adjacentes estão conectados por um polinômio de 3º grau, como apresentado na Equação 3.1. Então, derivando a Equação 3.1 duas vezes, e aplicando a representação do polinômio de interpolação de Lagrange, obtém-se a seguinte equação:

$$f_i''(x) = f_i''(x_{i-1}) \frac{x - x_i}{x_{i-1} - x_i} + f_i''(x_i) \frac{x - x_{i-1}}{x_i - x_{i-1}}. \quad (3.2)$$

onde $f_i(x)$ é o valor da segunda derivada em qualquer ponto x dentro do i -ésimo intervalo. Assim, esta equação é definida como uma reta que une a segunda derivada no primeiro nó $f_i(x_{i-1})$ com a segunda derivada no segundo nó $f_i(x_i)$.

A seguir, a Equação (3.2) se integra duas vezes para obter um expressão para $f_i(x)$, que é definida para um intervalo de pontos adjacentes, conforme a seguinte expressão:

$$f_i(x) = \frac{f_i''(x_{i-1})}{6(x_i - x_{i-1})}(x_i - x)^3 + \frac{f_i''(x_{i-1})}{6(x_i - x_{i-1})}(x - x_{i-1})^3 + \left[\frac{f(x_{i-1})}{x_i - x_{i-1}} - \frac{f''(x_{i-1})(x_i - x_{i-1})}{6} \right](x_i - x) + \left[\frac{f(x_i)}{x_i - x_{i-1}} - \frac{f''(x_i)(x_i - x_{i-1})}{6} \right](x - x_{i-1}). \quad (3.3)$$

Com relação à equação acima, os autores comentam que esta expressão conterá duas constantes de integração desconhecidas, que consistem nas segundas derivadas no início e final do intervalo ($f_i'(x_{i-1})$ e $f_i'(x_i)$).

Desta forma, para que a Equação 3.3 seja utilizada com um polinômio de 3º grau na interpolação de um intervalo de pontos, é necessário determinar as segundas derivadas nos nós. Assim, as derivadas são determinadas considerando a condição que as primeiras derivadas devem ser contínuas nos nós. Esta relação é estabelecida derivando a Equação 3.3 e igualando tanto para o intervalo $i-1$ como para o intervalo i , conforme apresentado na Equação 3.4.

$$(x_i - x_{i-1})f_i''(x_{i-1}) + 2(x_{i+1} - x_{i-1})f_i''(x_i) + (x_{i+1} - x_i)f_i''(x_{i+1}) = \frac{6}{(x_{i+1} - x_i)}[f(x_{i+1}) - f(x_i)] + \frac{6}{(x_i - x_{i-1})}[f(x_{i-1}) - f(x_i)]. \quad (3.4)$$

Por fim, escrevendo-se a Equação 3.4 para todos os nós internos, o resultado será $n-1$ equações distintas e um total de $n-1$ incógnitas, que podem ser determinadas a partir de um sistema de equações.

3.3.2 Controle dos módulos mecatrônicos

Com respeito ao controle de servoposicionadores pneumáticos, algumas estratégias de controle foram originalmente desenvolvidas para tais finalidades [Perondi, 2002; Sarmanho, 2014; Sobczyk, 2009].

Os algoritmos de controle destinados aos servoposicionadores podem ser aplicados para cada grau de liberdade do sistema modular, levando-se em conta que tais dispositivos são controlados através de um agente mecatrônico. Dado a que após a inicialização, o sistema Multiagente possui todas as informações cinemáticas e dinâmicas de cada dispositivo, e assim também, da estrutura robótica. Com isto define-se um modelo matemático que representa as características físicas do robô, facilitando a implementação de algoritmos de controle para o movimento das juntas do robô.

A Figura 3.14 ilustra um controlador realimentado aplicado a uma planta (robô modular). Para um caso de aplicação em um robô acionado pneumaticamente, são definidos alguns conceitos básicos:

Planta: Modelo matemático que representa a dinâmica dos servoposicionadores acoplados aos módulos mecatrônicos que compõe o robô;

Controlador: são os diferentes tipos de algoritmos que podem executar esta tarefa;

Referência: é a trajetória desejada que o sistema tem que realizar o seguimento.

Saída: é a posição do extremo dos atuadores em cada instante (se o objetivo é controlar a posição) a qual é comparada com a referência.



Figura 3.14 – Controle para cada módulo mecatrônico

Vale ressaltar que, segundo Perondi, 2002, para o controle de posição em sistemas pneumáticos, requer-se usualmente um tempo de ciclo de processamento do algoritmo de controle não ultrapasse um intervalo de 1ms.

A seguir, apresenta-se dois tipos de controladores clássicos que serão utilizados no sistema proposto: um controlador linear (PID), amplamente utilizado em sistemas industriais e um controlador não linear (SMC ou controlador por modos deslizantes), que pode operar com melhor desempenho em sistemas com variação de parâmetros e altamente não lineares, como é o caso dos sistemas pneumáticos.

Maiores detalhes sobre o desenvolvimento matemáticos destes controladores são descritos no Anexo A.

Controlador linear do tipo PID (Proporcional – Integral – Derivativo)

Este tipo controlador linear pode ser programado para efetuar tarefas de servoposicionamento pneumáticos.

A Figura 3.15 apresenta o algoritmo referente à lei de controle PID a ser implantada, onde, $u(t)$ é o sinal de controle, $e(t) = yd(t) - y(t)$ é erro de posição, Kp é o valor do ganho proporcional, Kd é o valor do ganho derivativo e Ki é o do ganho integral.

As variáveis de entrada do controlador PID são a referência em posição $y_d(t)$ e a posição efetiva do êmbolo $y(t)$. A diferença entre estas duas variáveis é processada pelos blocos proporcional, integrador e derivativo e seus respectivos ganhos.

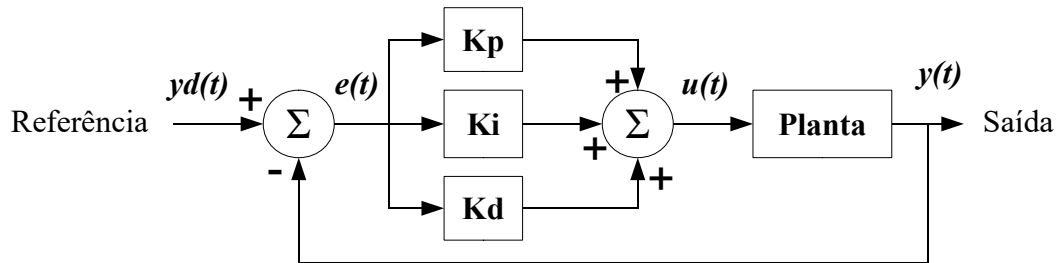


Figura 3.15 – Diagrama de Blocos PID.

Controle por modos deslizantes (*Slide Mode Control - SMC*)

Os controladores com estrutura variável são conhecidos pela alta robustez a variações paramétricas.

Com relação à utilização de controle por modos deslizantes em servoposicionadores pneumáticos, diversas pesquisas estudaram e propuseram este algoritmo de controle para tais aplicações, como, por exemplo, Golnaraghi e Kuo, 2009; Guenther e Perondi, 2000; Iordanou e Surgenor, 1997.

Estes controladores baseiam-se no chaveamento entre diferentes leis de controle a fim de manter os estados do sistema controlado dentro da chamada superfície de deslizamento S , que é especificada para um sistema operando em malha fechada.

Em uma situação ideal, uma vez que os estados do sistema atingem a superfície, eles mantêm-se no chamado regime de deslizamento, permanecendo em equilíbrio dinâmico ao longo da trajetória de estados definidos pela superfície (c_1 , c_2 e c_3). Desta forma, é necessária a aplicação de diferentes leis de controle para que a resposta se mantenha limitada pelo valor do erro especificado em projeto.

Em aplicações práticas, este chaveamento ocorre em um tempo finito, o que pode gerar oscilações (fenômeno conhecido por *chattering*) que, em última instância, pode acelerar o desgaste das peças móveis da servoválvula e ainda introduzir ruído no sistema Cukla, 2012.

Para a implantação deste controlador, a lei de controle por modos deslizantes é apresentada através da linguagem de blocos, conforme ilustrado na Figura 3.16.

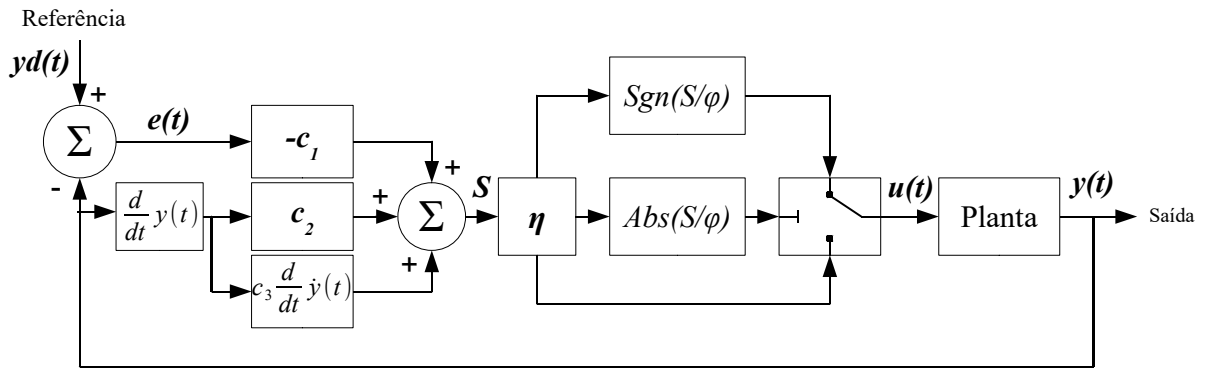


Figura 3.16 – Diagrama de blocos controle por modos deslizantes

De acordo com Slotine, 1988, para evitar o *chattering*, deve-se “suavizar” o sinal de controle $u(t)$, definindo uma camada limite φ dentro do qual ocorrerá a transição do sinal, conforme apresentado na Figura 3.16.

O termo $\eta > 0$ pode ser interpretado como uma medida da velocidade de deslocamento da trajetória em direção à superfície de deslizamento.

3.4 Os agentes e os módulos mecatrônicos

A seguir apresenta-se a proposta de uma arquitetura auto organizável, baseada em agentes, específica para a formação de robôs manipuladores de cadeias cinemáticas seriais. Ela se fundamenta na proposta apresentada por Cavalcante, 2012, em parte, na arquitetura CoBASA [Barata, 2003]. A presente proposta consiste de uma adaptação desses trabalhos para um sistema Multiagente utilizado para a montagem de robôs manipuladores.

3.4.1 Caracterização dos Módulos Mecatrônicos

Considerando o objetivo de criar uma arquitetura Multiagente aplicada a módulos mecatrônicos e com capacidade de auto-organização específica para um robô manipulador de uso industrial, e visando-se a um sistema adaptável, com capacidade de interação com o usuário e planejamento de trajetórias para determinadas operações, a solução proposta é a de uma arquitetura Multiagente em duas camadas, para o projeto dos agentes, e duas camadas de agentes executados dentro do próprio módulo mecatrônico, de forma a se ter uma arquitetura híbrida, conjuntamente deliberativa e reativa Wooldridge, 2009. A Figura 3.17 apresenta uma visão geral da solução proposta ao problema.

Como cada módulo mecatrônico está associado a um agente, define-se que este agente, pode ser interpretado como um sistema Multiagente com mais de uma entidade, enquanto de fora do módulo mecatrônico, é vista como uma única entidade com características mais complexas (por exemplo, módulo de acionamento linear 330mm, etc.). Assim, para estes módulos mecatrônicos, se propõe uma arquitetura que é formada por dois agentes com diferentes funcionalidades, instanciados conforme a necessidade, e nomeados de agente cognitivo e agente motor.

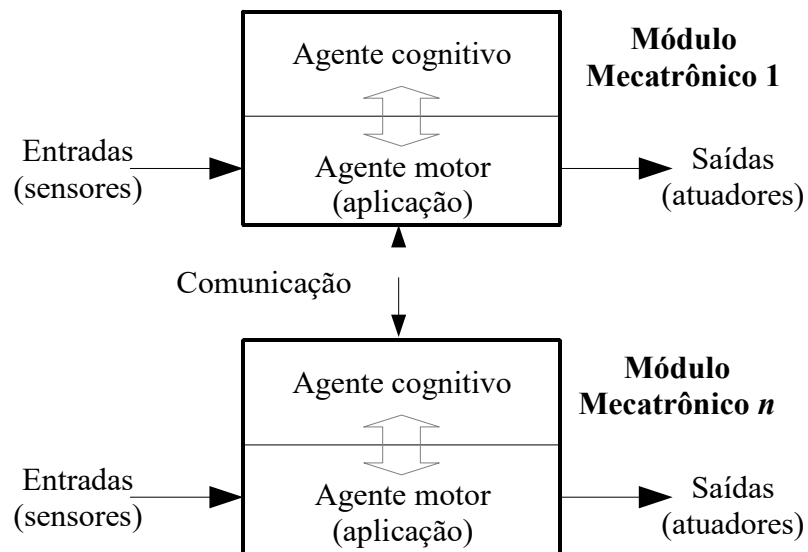


Figura 3.17 – Arquitetura proposta: visão geral [adaptado de Cavalcante, 2012].

A divisão em dois tipos de agentes se deve ao critério de funcionalidade: o agente cognitivo é responsável pela lógica da aplicação e integração dos agentes motores. Assim, o ambiente de um agente cognitivo é constituído por outros agentes cognitivos de funcionalidades menos complexas e, geralmente, agentes motores, que são os responsáveis pela ação no ambiente físico. Portanto, tem-se dois tipos de agentes; agente cognitivo e agente motor. Uma representação mais detalhada desta abordagem é apresentada na Figura 3.18.

A divisão em duas camadas internamente aos agentes segue o critério da comunicação: a camada superior do agente cognitivo é a aplicação propriamente dita do agente, enquanto sua camada inferior é responsável pela comunicação com outros agentes. No agente motor, a camada superior é a responsável pela comunicação e a camada inferior é a responsável pela aplicação ou interação com o meio físico.

Em um sistema assim baseado, onde um agente mecatrônico é formado por um agente cognitivo e um agente motor dentro de um mesmo módulo mecatrônico, assim cada agente

mecatrônicos pode se comunicar com outros módulos pares e compatíveis com a comunicação FIPA, tal como apresentado na Figura 3.18.

Assim, a camada inferior (de comunicação) no agente cognitivo tem dupla função: é responsável pelas mensagens de seleção do agente motor e pelas mensagens de requisição de ação a ele. A camada superior (de comunicação) no agente motor responde às mensagens de solicitação e de ação vindas do agente cognitivo.

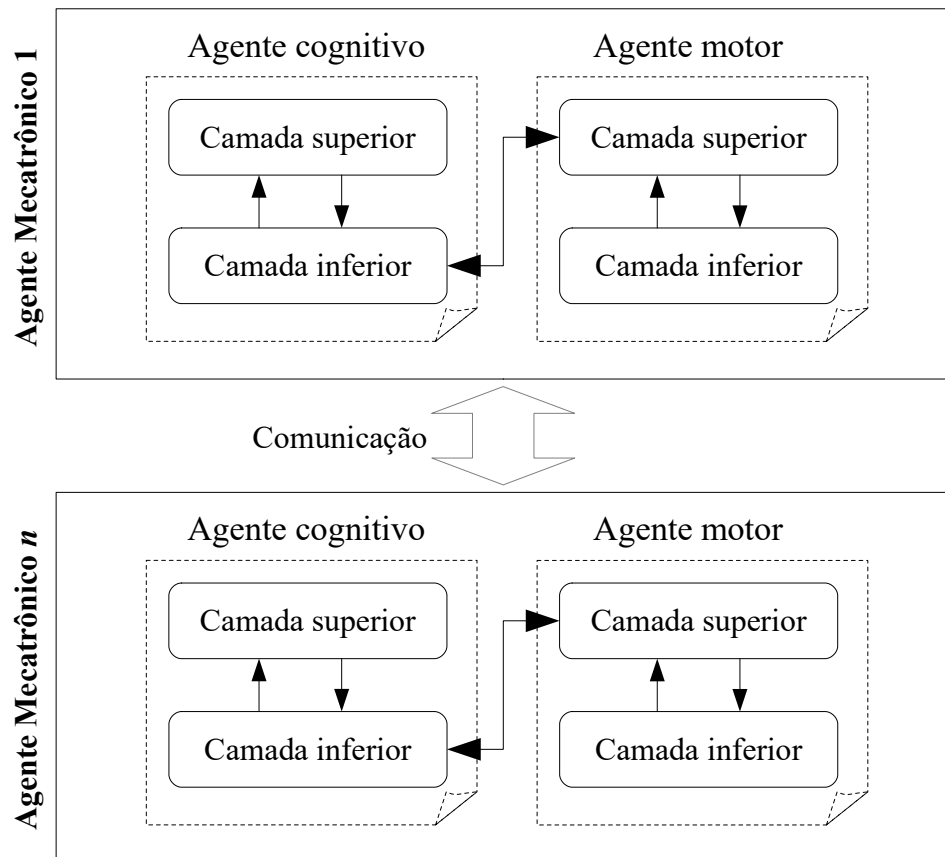


Figura 3.18 – Arquitetura proposta: cada agente cognitivo pode comunicar-se com vários agentes motores e outros agentes cognitivos.

A camada superior no agente cognitivo é a atividade fim do agente, isto é, a análise do ambiente (modelo matemático) e a interação com o agente motor (geralmente o do próprio módulo mecatrônico) mais adequado à realização de uma ação física no sistema.

No agente motor, a atividade fim é uma ação física do sistema, como as leituras de sensores e acionamentos de atuadores.

Para uma análise mais apropriada do ambiente, o agente cognitivo pode receber as informações e eventos do mundo físico percebidos pelo agente motor (por exemplo, a posição

do módulo no sistema). Isto acrescenta mais uma função às camadas de comunicação: enviar as percepções dos agentes motores aos agentes cognitivos.

A divisão em duas camadas de cada agente do sistema apresenta a vantagens de a separação de códigos, e através deste princípio, pode-se dividir um sistema em seções menores e mais maleáveis.

Cada seção de programa é responsável por uma única função (ou poucas funções). Assim, cada função permite o seu reuso em várias situações diferentes. Por outro lado, uma divisão muito intensiva se traduz em um aumento da complexidade do sistema, visto que as partes que o formam necessitam sempre trocar informações entre si. Isto geralmente redundando em perda de performance, mas em ganho de flexibilidade. Assim, optou-se pela menor divisão a fim de manter sistema o mais simples para implementação, sem grandes perdas de desempenho e em módulos com capacidades de processamento compatíveis aos requisitos. A divisão adotada, portanto, é de duas camadas.

Como a adoção de duas camadas propostas é baseada no critério da comunicação, todo o código do agente relacionado ao processo de comunicação vai para uma camada, deixando a outra como responsável pela aplicação propriamente dita. Essa abordagem em duas camadas permite a troca da tecnologia de comunicação sem grandes impactos na lógica da aplicação.

Comparando a um agente com código monolítico, cuja dependência da comunicação é completa, uma alteração qualquer na tecnologia de rede significa uma alteração no código da própria aplicação do agente. Em uma abordagem em duas camadas, uma alteração na tecnologia de rede pode ser feita sem afetar (ou afetar minimamente) a aplicação. Portanto, o uso de apenas duas camadas, divididas pelo critério de comunicação, tende a dar uma boa flexibilidade na criação de código dos agentes sem, contudo, penalizar excessivamente o desempenho da aplicação.

Outra questão que se coloca é sobre o uso de apenas dois tipos de agentes no sistema. Para analisar esta questão, usa-se um sistema formado por um único tipo de agente (um agente monolítico) responsável tanto pela parte cognitiva, quanto pela parte motora: neste caso, o agente carrega em si diversas responsabilidades, replicando a estrutura de um agente de raciocínio tradicional.

As limitações de performance podem ser importantes, pois, para qualquer ação, uma análise (às vezes profunda) se faz necessária.

A proposta apresentada divide o problema em dois subproblemas: as atividades que requeiram maior deliberação vão para o agente cognitivo, e as atividades de acesso ao hardware (e.g. sensoriamento e atuação) vão para o agente motor.

Por outro lado, se mais tipos de agentes fossem propostos, dividindo-se ainda mais as funções dos agentes além da cognição e da interação com o médio (atuação e sensoramento), haveria um aumento inerente na complexidade do sistema.

Neste caso, há um custo maior de programação ocasionado pela necessidade de mais tipos de interações diferentes a serem criadas em virtude de um maior número de tipos de agentes. Em muitos casos, esse aumento da complexidade pode inviabilizar o uso em sistemas mínimos.

Uma aplicação desta arquitetura proposta é a montagem de um manipulador modular, que é formado por módulos mecatrônicos, onde um agente representa a cada módulo mecatrônico. Estes agentes mecatrônicos, estão compostos por agentes cognitivos e agentes motores, tal como apresentados na Figura 3.19.

Os agentes cognitivos, modelam a dinâmica do robô e executam os algoritmos de controle, enquanto que os agentes motores gerenciam os recursos de interação com o meio e executam funções de atuação e leituras de sensores. Neste tipo de aplicação, os agentes cognitivos executam processos e os agentes motores executam o acesso ao hardware.

Com a proposta apresentada pode-se realizar ambas abordagens, deliberativa e reativa, eliminando-se as limitações de cada modelo.

O agente cognitivo é mais próximo a um agente de raciocínio tradicional e um agente motor é mais próximo a um agente reativo. Os agentes cognitivos e motores, compõem o agente mecatrônicos, e são executados em um módulo mecatrônico.

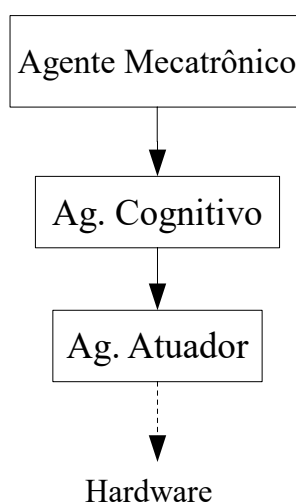


Figura 3.19 – Agentes que compõem um módulo mecatrônico

Execução de funcionalidades dos agentes mecatrônicos

Em uma estrutura modular, cada agente, associado a um módulo mecatrônico, é responsável pela execução de uma única ou poucas atividades no processo de movimentação do braço. Tais atividades são as funcionalidades do sistema.

Cada agente mecatrônico, portanto, possui um conjunto de funcionalidades ou habilidades e é capaz de executá-las sob requisição externa ou por decisão interna. Então, uma funcionalidade é uma entidade que encapsula as informações que definem uma ação específica de um agente mecatrônico no sistema. Essa ação pode ser tanto física quanto lógica, por exemplo, o acionamento de uma válvula ou a execução de uma expressão matemática, ou ainda, a execução de um algoritmo de controle. As funcionalidades estão definidas em todos os agentes mecatrônicos envolvidos, e o objetivo final do agente é a sua execução.

Há duas formas de um agente iniciar a execução de uma funcionalidade sua: quando for solicitado, ou por auto execução, em geral na inicialização do agente. Um processo é um conjunto de “chamadas” a outras funcionalidades, agrupadas de uma forma lógica, por exemplo, mover o braço de um robô ou acionar o efetuador. Uma chamada a uma funcionalidade nada mais é que a requisição da execução daquela funcionalidade pelo agente que contém o processo a um agente capaz de executar tal funcionalidade.

Quando um agente for realizar um determinado processo, a chamada para executar dita funcionalidade pode ser local ou remota, isto é, a requisição para sua execução é do próprio agente mecatrônico que executa o processo ou é de um outro módulo mecatrônico. Então, a requisição de execução de uma funcionalidade é equivalente a uma chamada de um procedimento local ou remoto pelo agente que contém o processo.

Uma chamada de procedimento local é uma chamada a uma função escrita em uma linguagem de programação qualquer (e.g. método nativo para linguagens orientadas a objetos, como Java ou C++) que está no mesmo espaço de endereçamento do requisitante. Uma chamada de procedimento remoto é uma chamada de função em outro espaço de endereçamento, em geral, em outro módulo mecatrônico.

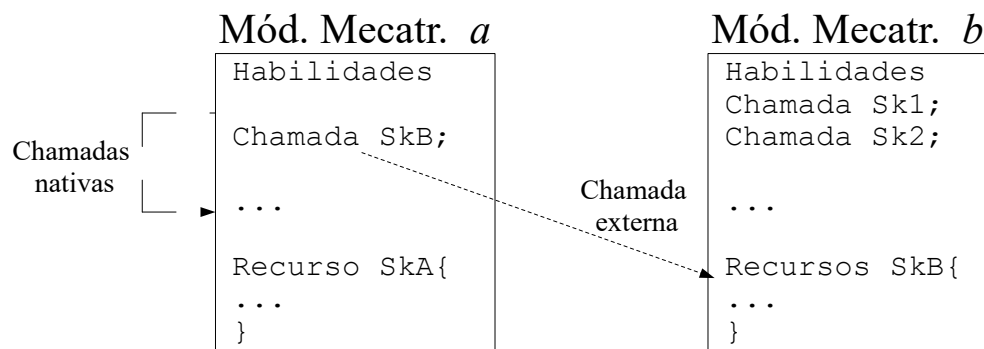


Figura 3.20 – Exemplo de funcionalidades em agentes cognitivo e motor

A Figura 3.20 mostra como dois módulos mecatrônicos interatuam. Assim, dentro de cada módulo, há um agente cognitivo, que é um processo que define funcionalidade ou “habilidades” do sistema, e um agente motor, chamado “recursos”. Cada uma dessas funcionalidades pode ser executada mediante um procedimento nativo, ou chamadas externas. As chamadas locais são feitas diretamente, ao agente motor. Já, as chamadas externas são requisitadas ao agente mecatrônico responsável pela execução de um determinado recurso.

Assim, conforme já discutido, a proposta apresenta dois tipos de agentes: um agente cognitivo e um agente motor. A diferença entre eles é que o agente cognitivo permite a execução de funcionalidades como um processo, isto é, cada uma de suas funcionalidades são executadas como uma série estruturada de chamadas de outras funcionalidades, que podem ser chamadas externas, enquanto o agente motor executa funcionalidades, como as chamadas de procedimentos locais.

Uma alternativa para um processo em um agente cognitivo é a divisão em blocos funcionais ou funções de processos. As funcionalidades do agente cognitivo são executadas como uma função do tempo que é periodicamente chamada (os processos de controle). Na Figura 3.21 é mostrado um exemplo de chamada de funcionalidades entre dois agentes cognitivos, onde o primeiro agente (Modelo) que implementa um bloco funcional faz uma chamada em outro agente (Controle). Isto pode ser usado para implementar um sistema de controle distribuído.

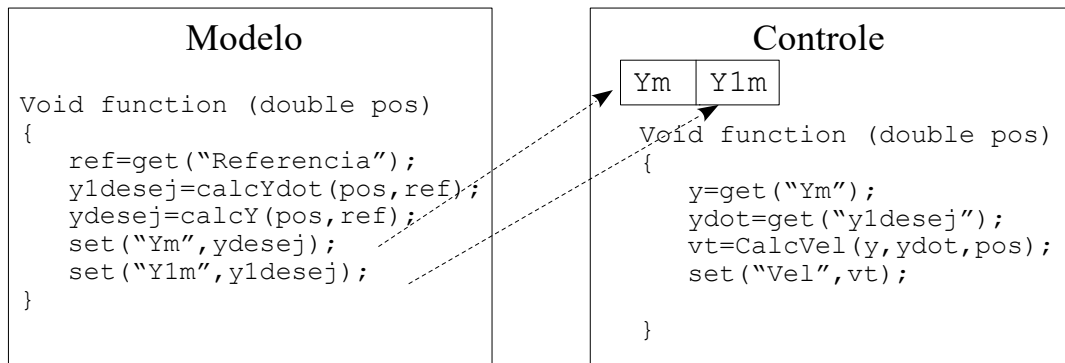


Figura 3.21 – Agente cognitivo implementando um bloco funcional

Neste caso, é importante destacar que o agente Modelo pode ser considerado um agente de um nível hierárquico maior ao agente Controle, onde o agente cognitivo Modelo, pode ser uma abstração matemática de todo um sistema, e realiza as chamadas de funcionalidades a entidades de mais baixo nível, como, por exemplo, no caso, um robô modular.

Por outro lado, um agente motor apenas executa uma chamada de procedimento local, o qual é, em geral, estático e definido em tempo de projeto, e sempre depende de um agente cognitivo.

3.4.2 Definição da Estrutura Robótica

Uma estrutura robotizada baseada na composição de módulos mecatrônicos, deve necessariamente apresentar uma organização a nível lógico bem definida, isto é, deve estabelecerem como são organizados os agentes na formação de um robô manipulador. Desta forma, a seguir, define-se como estes agentes interagem dentro de uma já estrutura formada, e como são administradas as suas habilidades para o gerenciamento de um robô manipulador e os diferentes níveis hierárquicos dos agentes.

Formação de estruturas robotizadas a partir da coalisão de módulos mecatrônicos

A arquitetura proposta tenta apresentar uma estrutura organizacional do sistema de forma que seja menos complexa possível, através de uma hierarquia de agentes. Como exemplo, na Figura 3.22 é apresentada uma estrutura de um sistema modular com três níveis hierárquicos. O agente cognitivo “Cognitivo1” executa remotamente as funcionalidades nos agentes “Atuador1”, “Atuador2” e “Cognitivo2”. Do seu ponto de vista, todas as chamadas são atômicas, isto é, indivisíveis e são simplesmente chamadas às funcionalidades remotas.

Entretanto, o agente “Cognitivo2” implementa a sua funcionalidade como um processo, isto é, através de um diagrama de blocos que representa o conjunto de suas atividades, que, por sua vez, encapsulam as chamadas aos agentes “Atuador3” e “Atuador4”. Note-se que o agente “Cognitivo1” não percebe a existência de “Atuador3” e “Atuador4” no sistema.

Dessa forma, em um sistema complexo que possua vários módulos mecatrônicos, estes podem ser colocados juntos para formar uma estrutura mecatrônica maior. Neste caso, o controlador de cada subcomponente de um módulo mecatrônico (por exemplo os sensores ou atuadores) pode ser implementado como um agente motor, que externa algumas funcionalidades, as quais são agregadas em uma única interface através de agentes cognitivos, que, por sua vez, externa as funcionalidades do módulo complexo ao restante do sistema.

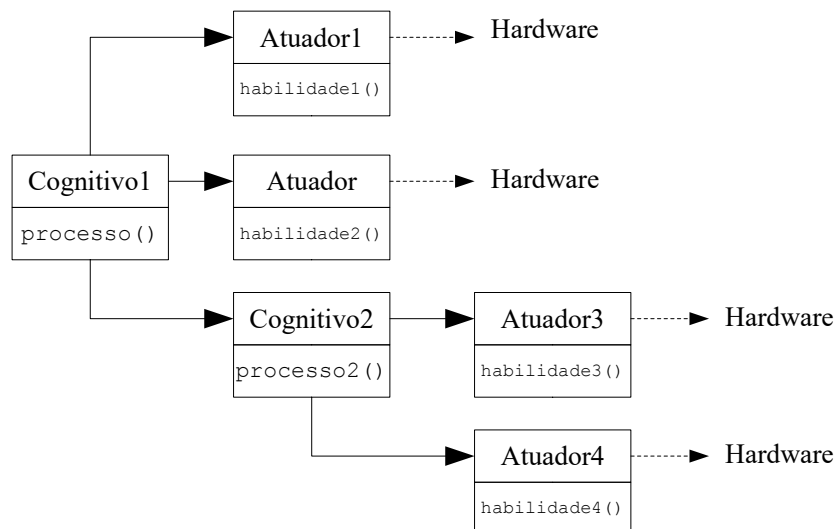


Figura 3.22 – Hierarquia de agentes mecatrônicos chamando funcionalidades remotas.

Um exemplo típico de tal arranjo é o caso de um manipulador de três graus de liberdade do tipo cartesiano (Figura 3.23), onde cada elo é um módulo mecatrônico, o qual possui um agente motor e um agente cognitivo associado a cada elo, e, assim também, sistema implementado é visto de fora como um agente cognitivo que faz uso de chamadas aos agentes responsáveis pelas movimentações de cada articulação.

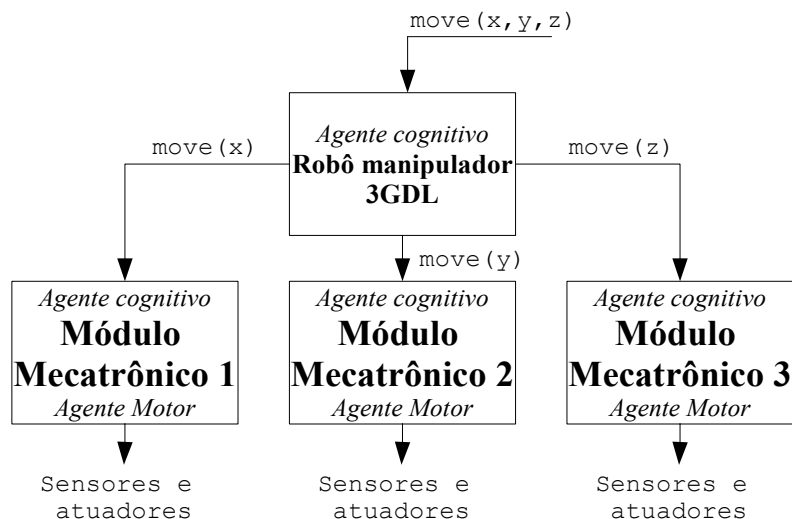


Figura 3.23 – Coalizão de agentes mecatrônicos em um manipulador de 3GDL

Assim, a proposta aqui apresentada permite interligar módulos mecatrônicos em vários níveis, ocultando ou não a sua complexidade, de tal modo a favorecer a modularidade e flexibilidade do sistema e a interação com outros sistemas complexos de forma simplificada, tal como são as linhas de produção.

Reconhecimento das características do manipulador modular

Na formação de robôs manipuladores, foram definidos alguns elementos típicos: Módulos Mecatrônicos de acionamento linear, Módulos Mecatrônicos de acionamento rotacional e efetadores (garras). Estes elementos podem ser conectados formando uma determinada estrutura cinemática. É possível mapear estes elementos e determinar a posição de cada elemento dentro da cadeia cinemática em que encontra. Para testar, após a montagem de uma estrutura cinemática, cada módulo mecadrônico é fixado e alimentado mediante conexões específicas (Apresentado na Seção 3.1). Nesta seção, é descrita como são inicializadas as funcionalidades de um agente mecadrônico, sua posição hierárquica na estrutura de agentes, a sua capacidade de receber solicitações e, efetivamente, executar as funcionalidades requisitadas.

Estas funcionalidades são de dois tipos: execução de processos (ex. planejamento de trajetória, processamento de um algoritmo de controle, etc.) e execução de procedimentos locais (acesso ao hardware: leitura de sensores e comando dos atuadores). Com essas funcionalidades, é possível então diferenciar dois tipos de agentes que a executam: agentes cognitivos (que

realizam a execução de funcionalidades como um processo) e agentes motores (que realizam o acesso ao hardware).

Quando o mapeamento de elementos ocorre a nível macroscópico, por exemplo, em linhas de manufatura que utilizam sistemas Multiagentes, o mapeamento dos elementos físicos e sistemas mecatrônicos podem ser classificados em módulos simples e subsistemas de maiores complexidades. Um módulo mecatrônico é mapeado como um agente com uma funcionalidade específica (funcionalidade atômica). Este módulo mecatrônico com uma única funcionalidade (deslocamento linear), está apresentado na Figura 3.24 e recebe o nome genérico de agente recurso. Os agentes recursos são particularmente associados a sistemas físicos (como um servoposicionador pneumático), já que são eles os que realizam o acesso ao hardware.

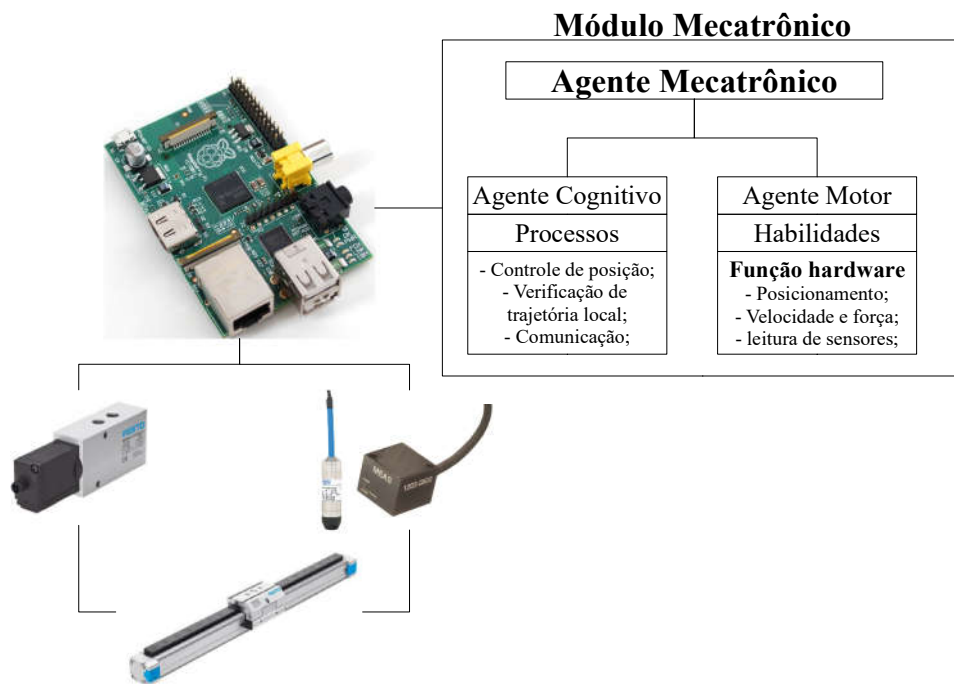


Figura 3.24 – Mapeamento de módulo mecatrônico como agente simples (servoposicionador linear)

Um módulo mecatrônico, seja ele de acionamento linear ou rotacional, ou mesmo um efetuator, é visto como uma entidade que dispõe de uma certa funcionalidade e de acesso ao hardware. Cada módulo mecatrônico que realiza uma função específica tem características físicas próprias das quais os agentes dispõem das informações técnicas específicas (peso, dimensões, comprimento do percurso, velocidade máxima, etc.).

Já, um subsistema complexo é um arranjo de módulos mecatrônicos, tal como é um robô modular, ou uma estação de trabalho com mais de uma funcionalidade. Cada sistema formado

por um ou mais módulos mecatrônicos pode ser mapeado por um agente cognitivo, o qual irá realizar a coalizão dos agentes de recursos responsável pelos módulos individuais. Este agente recebe o nome de agente líder de coalizão e representa um sistema mais complexo. Assim, este agente é visto de fora como um único agente com funcionalidades complexas (por exemplo um robô manipulador). Na Figura 3.25 está apresentado este tipo de agrupação.

Agentes líderes de coalizão, que são agentes cognitivos, são particularmente aplicados para estruturar hierarquias de agentes e módulos no sistema.

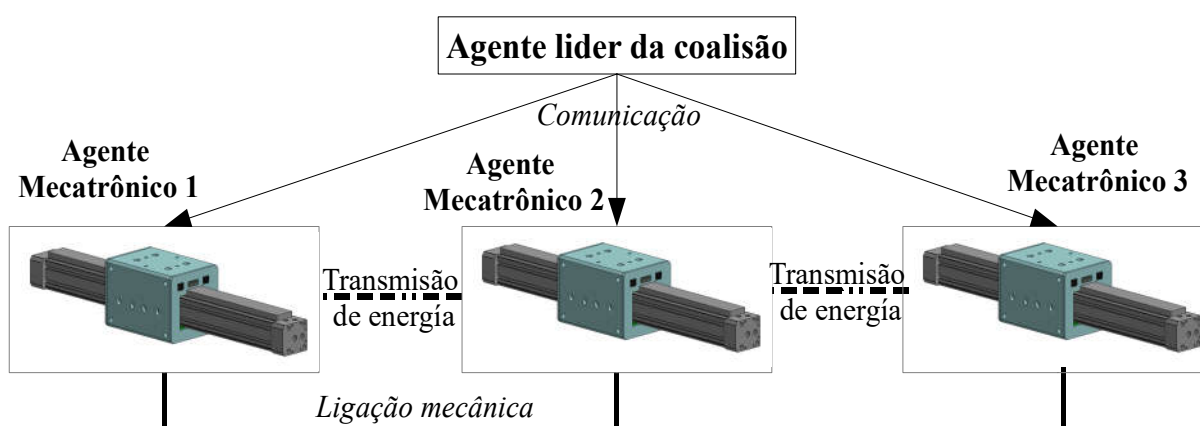


Figura 3.25 – Cada módulo mecatrônico do sistema pode ser mapeado diretamente a um agente motor

Cada módulo mecatrônico é reconhecido representando por um único agente que dispõe de certas funcionalidades específicas. Assim, um agente líder representa uma estrutura mais complexa, formada pela coalizão destes módulos mecatrônicos. As suas funcionalidades são definidas pelas habilidades herdadas dos agentes individuais. O agente líder, não tem definido um lugar físico de alocação, tal como é o caso de dos agentes mecatrônicos. Desta forma, o agente líder é inicializado no primeiro módulo mecatrônico que é ativo no sistema, e compartilha o mesmo sistema de processamento do agente mecatrônicos.

A nomenclatura utilizada neste trabalho, tais como agente recurso, agente líder de coalizão, agente motor e agente cognitivo, é derivada da arquitetura CoBASA [Barata, 2003] e usada no IADE [Ribeiro et al., 2012] e o trabalho de tese de [Cavalcante, 2012]. Tanto nesses trabalhos como nesta pesquisa, usam-se os termos agente recurso para um agente motor e agente líder de coalizão para um agente cognitivo.

Resposta de auto-organização e serviços de páginas amarelas

Além dos agentes já mencionados (agente líder de coalisão, agentes mecatrônicos, agentes motores e agentes cognitivos), a proposta apresentada insere um outro tipo de agente: o agente de páginas amarelas.

A fim de que um agente mecatrônico possa divulgar as suas funcionalidades (tipo de acionamento, peso, velocidade máxima, força máxima, etc.) para que os demais agentes mecatrônicos possam acessá-las, o sistema deve contar com um serviço de publicação e descoberta de funcionalidades. Portanto, é necessário que um agente do sistema providencie serviços comumente denominados de “serviços de registro” ou “serviços de páginas amarelas” [Wooldridge, 2009].

Há três serviços básicos em um agente de páginas amarelas:

- (i) um serviço de registro de funcionalidades por um agente;
- (ii) um serviço de busca de funcionalidades de um agente; e
- (iii) um serviço de busca de agentes que podem executar uma dada funcionalidade.

Logo na sua iniciação, todos os agentes devem registrar suas funcionalidades no serviço de páginas amarelas, a fim de disponibilizá-las para os outros agentes. Posteriormente, quando um agente cognitivo necessitar saber quais agentes são capazes de executar uma dada funcionalidade, utiliza o serviço de busca apropriado.

A busca por agentes com determinadas funcionalidades ocorre no processo de negociação, utilizando o protocolo FIPA para a comunicação mencionado no Capítulo 2. O agente iniciador da conversação deve enviar a mensagem (*Call for Proposal*) CFP para todos os participantes que podem executar aquela funcionalidade. Para fazer isso, o agente deve, primeiro, buscar no serviço de páginas amarelas os agentes que podem executar a funcionalidade. Os agentes com capacidades ou funcionalidades, para fins de registro ou busca, é designado um identificador único no sistema para cada um deles.

Além destes, outro serviço útil é o serviço de registro do agente, para realizar sua limpeza quando o agente é destruído. Entretanto, o próprio agente de páginas amarelas pode realizar este serviço de limpeza automática se houver meios de detectar a saída do agente do sistema.

O agente de serviços de páginas amarelas realiza uma restrição lógica no sistema: para cada agente de páginas amarelas, somente o grupo de agentes mecatrônicos que realizou o seu registro está no escopo de buscas naquele agente de serviços de páginas amarelas. Essa restrição lógica, em geral, modela também uma restrição física. Por exemplo: em um sistema de três

eixos não há necessidade de o agente de coalizão realizar uma busca além dos agentes dos eixos fisicamente presentes em uma área restrita. Então, pode-se criar um agente de páginas amarelas apenas para aquele subconjunto de agentes mecatrônicos: os três agentes recursos e o agente líder de coalizão, tal como apresentado na Figura 3.26.

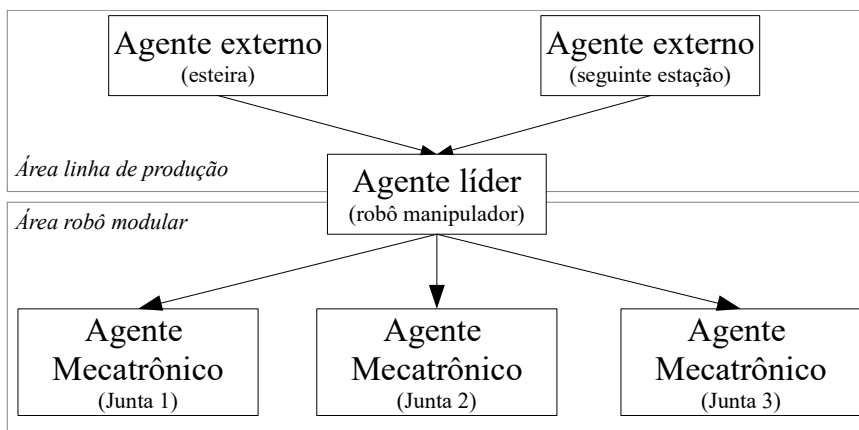


Figura 3.26 – Restrição de área para a interação dos agentes e formação de pagina amarela.

Diz-se, então, que um agente de páginas amarelas define uma área de atuação. Daí o sistema de maior complexidade, como é o caso de uma linha de produção, terá tantas áreas quanto o número de agentes de páginas amarelas.

Assim, da mesma forma que os módulos mecatrônicos publicam as suas funcionalidades na página amarela referente a “área robô modular”, o robô manipulador tem a necessidade de publicar as suas capacidades em uma página amarela de uma área de maior abrangência, chamada aqui de linha de produção, na qual todas as estações de trabalhos representados por agentes externam seus serviços.

Essa situação é mostrada na Figura 3.26, onde os Agentes Mecatrônicos são recursos, e os demais (estação de trabalho, esteira e robô manipulador) são líderes de coalizão. Neste caso, o agente líder “robô manipulador” deve se inscrever tanto no agente de páginas amarelas da “Área linha de produção”, a fim de disponibilizar serviços, quanto no da “Área robô modular”, e poder pesquisar agentes mecatrônicos. Esta abordagem de maior abrangência não será tratada neste trabalho.

Além disso, nota-se que a noção de área gera uma restrição na quantidade de agentes mecatrônicos que podem responder a um evento de auto-organização (negociação e execução de funcionalidades), isto é, a área restringe a resposta de auto-organização do sistema.

O agente de páginas amarelas não é um agente mecatrônico, pois não possui funcionalidades, seja para realizar acesso ao hardware, seja para a execução de processos, formando uma outra classe de agente que deve ser, necessariamente, adicionada à arquitetura proposta.

Agente de distribuição

As descrições das funcionalidades são como uma “língua franca” entre os agentes. Elas devem ser publicadas e pesquisadas em serviços de páginas amarelas, devem ser trocadas entre agentes para negociação ou execução de suas funcionalidades. A descrição de uma funcionalidade deve ser de uma forma tal que possa ser usada durante um processo de comunicação entre agentes. Diz-se então que uma funcionalidade tem uma forma serializada. A definição de uma forma serializada para funcionalidades permite a criação de uma forma serializada também para os agentes mecatrônicos. Tal forma teria, no mínimo, a identificação única do agente e uma lista de suas funcionalidades em suas formas serializadas.

Isto permite fazer a carga dessas funcionalidades dinamicamente em cada agente. Neste caso, o código de cada entidade passa a ser genérico, de forma que pode executar qualquer tipo de funcionalidade. Então, a única diferença entre os agentes cognitivos e motores reside na definição do motor de execução de suas funcionalidades. Em um agente cognitivo, um motor de execução é criado para executar funcionalidades descritas como um processo, ou como um bloco funcional. Em um agente motor, a funcionalidade encapsula a execução de uma chamada de procedimento nativo.

A existência de uma forma serializada para um agente mecatrônico torna possível a sua instanciação sob demanda. Para isso, um agente especial, o agente de distribuição, toma essa forma serializada e a instancia na máquina local onde é executado. Através do agente de distribuição, consegue-se uma espécie de mobilidade de código, o que torna o sistema efetivamente dinâmico.

Os formatos serializados de agentes mecatrônicos, de funcionalidades e a existência de um agente de distribuição, torna possível o desenvolvimento de uma ferramenta de criação, distribuição e monitoração de agentes para o sistema. Da mesma forma que o agente de páginas amarelas, o agente de distribuição não é um agente mecatrônico, mas um outro agente que deve ser inserido na arquitetura para permitir a instanciação de agentes mecatrônicos sob demanda.

3.4.3 Agentes da plataforma

De acordo com as características dos agentes até agora especificadas, assim como a suas variedades, a seguir propõe-se um modelo conceitual a partir de uma visão orientada a objetos e mostrada na Figura 3.27.

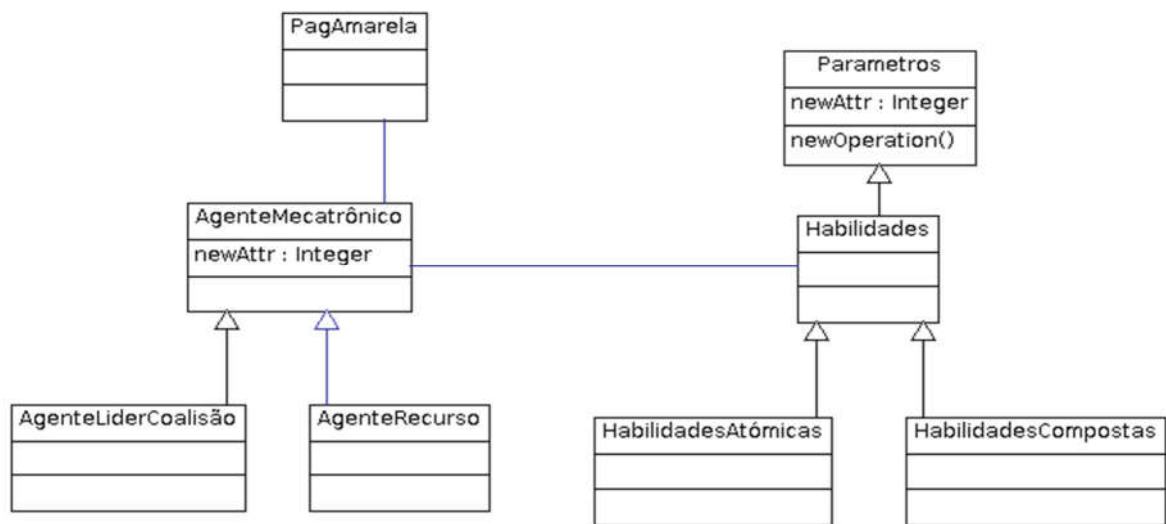


Figura 3.27 – Diagrama orientado a objetos dos agentes

Estas definições serão a base de uma plataforma para criação de robôs modulares baseados em módulos mecatrônicos.

Desta forma, a proposta apresentada neste trabalho, pode ser resumida neste diagrama, onde *AgenteRecurso* é um agente motor, já que realiza acesso ao hardware, já o *AgenteMecatrônico* é quem possui uma funcionalidade atômica (agente cognitivo), enquanto *AgenteLiderCoalisão* é um agente cognitivo porque executam funcionalidades e funções de alto nível. Por outro lado, há agentes que não são mecatrônicos, mas fornecem serviços extras e fundamentais para a plataforma. Estes agentes estão descritos a seguir.

Definição dos Agentes Mecatrônicos

Dentre os agentes com capacidade de cognitivas ou capacidades motoras, é possível definir os três principais que envolvem este trabalho:

O Agente Recurso (RA) ou *AgenteRecurso* é um agente motor e representa uma função física, isto é, externa os serviços que são executados pelo módulo mecatrônico. Portanto, o

comportamento global de um RA inclui os aspectos de atuação e leitura do equipamento. Um agente Recurso sempre é processado pelo módulo mecatrônico que ele o representa.

Já por meio do Agente Líder de Coalisão (CLA) ou *AgenteLiderCoalisão* são representadas as funcionalidades do robô modular. Este agente é conformado pela coalisão dos outros agentes do sistema. Esta entidade é responsável pela agregação e orquestração dos agentes e funcionalidades do sistema, em particular dos RAs. Um CLA suporta a execução lógica de um processo complexo a ser executado no sistema. É o CLA que executa negociação com outros agentes para a execução das funcionalidades disponíveis e expõe, por si, uma funcionalidade ao sistema.

O Agente Mecatrônico ou *AgenteMecatrônico* é o agente base da plataforma, e é executado no seu respectivo módulo mecatrônico. É o agente mecatrônico que apresenta características cognitivas de acesso ao hardware (tal como ilustrado na Figura 2.13). Um agente mecatrônico possui um identificador único e um conjunto de funcionalidades (habilidades). O diagrama da Figura 3.27 representa uma lista de elementos, onde nenhuma instância de *AgenteMecatrônico* pode ser criada, necessitando-se a criação de alguns descendentes, os quais de fato serão os agentes do sistema.

Como a diferença entre um agente cognitivo e um agente motor reside no motor de execução de suas funcionalidades, então um *AgenteLiderCoalisão* é um tipo de *AgenteMecatrônico* que provê a implementação do motor de execução de um plano de processos complexos descrito como grafo de funcionalidades compostas. Já um *AgenteRecurso* é um *AgenteMecatrônico* que provê a execução das funcionalidades como chamadas de método nativos.

Cabe às classes descendentes de *AgenteMecatrônico* a definição das formas organizadas dos respectivos agentes, as quais serão usadas pelo *AgenteDistribuição* para a instanciação sob demanda daquele agente.

Definição das habilidades dos agentes

As funcionalidades, por sua vez, são representadas no diagrama da Figura 3.27 por uma classe *Habilidades*, que define um nome e os parâmetros da funcionalidade. Tal classe também é abstrata pois define apenas as propriedades gerais para funcionalidades. Cabe aos descendentes as definições específicas. Então, *HabilidadesAtômicas* define uma chamada de procedimento nativo. Um processo é definido por *HabilidadesCompostas*, que permite o

agrupamento de outras funcionalidades. Cabe também aos descendentes de *Habilidades* a definição das formas serializáveis das funcionalidades.

Uma habilidade é, pois, uma funcionalidade específica dentro da arquitetura, a qual é disponibilizada por um CLA ou RA. Todas as funcionalidades externadas por agentes mecatrônicos tomam esta forma. São distinguíveis dois tipos: habilidades atômicas e habilidades compostas.

As *Habilidades Atômicas (ASk)* são integradas a um RA, possuindo os detalhes técnicos de interação e integração com o equipamento concreto sendo controlado. Em particular, ele encapsula uma instância da biblioteca de baixo nível (o nível do controle de E/S) e mapeia funções de baixo nível com as funções de ordem mais alta ao nível do agente.

Assim, as *Habilidades Composto (CSk)* as quais são integradas a um CLA, suportam o projeto do processo de coalizão de Habilidades. É chamado de composto, porque admite a inserção de sub habilidades, incluindo outros CSk.

Uma Habilidade especial, somente é executada localmente quando da execução de um plano de processo em um CLA, é uma Habilidade de Decisão (DSk) usada para resolver, em tempo de execução, expressões booleanas e executar outras habilidades do plano de forma condicional.

Definição dos demais agentes

Os outros agentes não mecatrônicos são: Agente Pagina Amarela e Agente de Distribuição.

O *AgentePagAmarela (YPA)*, é um agente que externa serviços especializados de páginas amarelas para as funcionalidades e demais agentes na plataforma. Cada plataforma deve possuir ao menos um agente de páginas amarelas, instanciado dentro do contenedor principal. Cada YPA extra delimita um escopo de busca para agentes e um contexto específico para CLAs e RAs.

Já, o *AgenteDistribuição (DA)*, é um agente especializado em receber e instanciar uma versão serializada dos agentes principais da plataforma. É interessante notar que este é, de fato, o único agente que deve ser diretamente instanciado no módulo mecatrônico onde a plataforma será executada, pois, a partir dos DAs, os CLAs e RAs são instanciados.

O modelo da Figura 3.27, é o conjunto mínimo de agentes e classes que define a plataforma. As implementações irão definir mais campos e classes de apoio, bem como ferramentas para criação e visualização das formas serializadas.

Note-se que, por causa das formas serializadas dos agentes, para um sistema iniciar, somente é preciso um único agente de páginas amarelas executando em alguma CPU, um agente de distribuição executando em cada CPU que se queira instanciar agentes, e um agente gerador das formas serializadas dos demais tipos de agentes, para se ter um sistema pronto para operar.

Uma visão das relações entre os agentes até agora discutidos, pode ser visualizada no diagrama da Figura 3.28. Tal modelo possui as condições para que o sistema Multiagente seja capaz de prover a escalabilidade (inserir e retirar elementos modulares), através da entrada e saída de agentes mecatrônicos no sistema, reagindo de imediato a essas ocorrências [Ferreira et al., 2012].

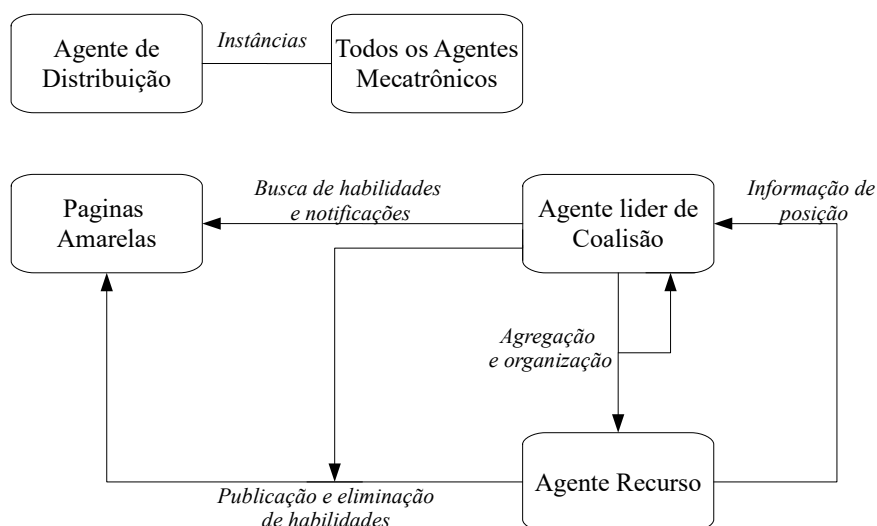


Figura 3.28 – Os agentes da arquitetura e seus relacionamentos.

3.5 Considerações finais

Quanto aos aspectos relativos ao projeto mecânico dos robôs modulares, mesmo que o presente trabalho não aborda inteiramente as etapas de projeto e fabricação mecânica dos seus elementos (materiais, dimensionamento dos elementos de acionamento, tipos de conectores, etc.), são especificadas características como o tipo de fixação dos componentes, as formas de acionamento, o modo de acoplamento, os elementos de acionamento, entre outras, as quais são importantes para estabelecer as etapas de montagem e as funcionalidades deste tipo de robô. Nesse sentido, o presente estudo apresenta uma abordagem geral para a construção de manipuladores de uso industrial a partir de módulos mecatrônicos, já que o foco desta pesquisa se baseia em uma proposta para definição dos módulos mecatrônicos na formação de robôs

modulares, de modo que os aspectos de projeto das unidades mecatrônicas deverão ser abordados em futuras pesquisas.

Em relação à montagem mecânica, define-se que cada conexão mecânica, por meio dos módulos auxiliares, apresenta duas possibilidades de fixação entre dois módulos mecatrônicos, permitindo, assim, diferentes orientações entre os mesmos. Este tipo de solução mostra-se apropriada para sistemas de acionamento linear. Para sistemas de acionamento rotacional são necessárias algumas considerações quanto à forma de conexão entre os módulos, as quais não foram abordados diretamente no presente trabalho.

O sistema elétrico, cujos circuitos foram utilizados no protótipo de um manipulador cartesiano de acionamento pneumático, mostra-se também apropriado para robôs modulares com acionamento pneumático, visto que as características técnicas da plataforma elétrica proposta atendem os requisitos de aquisição, atuação e processamento de dados, que são premissas necessárias nas aplicações de sistemas pneumáticos em robótica. Outro fator importante consiste no fato de que o sistema proposto, na maior parte do seu tratamento de dados, trabalha com sinais digitais, os quais apresentam maior imunidade ao ruído e facilidade de manutenção. Além disso, este sistema elétrico possui uma interface de comunicação que utiliza a rede local ethernet (conexão via cabo por meio de conectores do tipo RJ45) viabilizado uma troca eficiente de informações entre os agentes. A rede proposta, em função do espaço necessário para o cabeamento, pode apresentar inconvenientes, como, por exemplo, surgimento de ruído tribológico nas movimentações relativas entre as juntas. Estas questões podem ser solucionadas mediante a utilização de uma rede WiFi em cada uma das unidades mecatrônicas, que deverá ser objeto também de trabalhos futuros.

O conceito de sistema Multiagente abordado neste trabalho, proporciona uma abordagem de unidades independentes e com capacidade de trabalho em conjunto de todos os módulos mecatrônicos que é refletida na correta movimentação de um robô manipulador modular. O sistema Multiagente, através dos seus comportamentos e a comunicação entre as entidades, têm a capacidade de executar todas as funções necessárias para o correto funcionamento do robô, tais como a execução dos algoritmos de controle, reconhecimento da estrutura, planejamento de trajetória e interface com o usuário para a inserção dos pontos intermediários.

Os primeiros resultados das simulações mostram que os sistemas Multiagentes são uma técnica promissora para a montagem de sistemas modulares, considerando diferentes restrições de operação. Ainda são necessários vários tipos de implantações no sistema Multiagentes, tal como a otimização de intercâmbio de mensagens entre as entidades, amplo detalhamento de

todas as atividades que cada unidade deve realizar (o que pode implicar em um maior tempo de processamento de dados e comunicação entre os agentes). Sem dúvida, a simulação apresenta uma situação simplificada e controlada dos comportamentos agentes mecatrônicos. Por outra parte, as plataformas de agentes foram testadas em execuções de três tarefas básicas: reconhecimento da estrutura formada, cálculos da cinemática e controle de posicionamento dos módulos mecatrônicos.

4 ROBÔS MODULARES

Os módulos mecatrônicos, apresentados no Capítulo 3 são unidades que podem ser acopladas na formação de um robô manipulador. Dependendo do tipo de elementos utilizados e a forma de conexão, a combinação adequada destes permite a montagem de robôs cilíndricos, cartesianos, antropomorfos, etc. Neste estudo, levando em conta trabalhos anteriores focados na formação de robôs modulares [Chen e Yang, 1996; Larizza et al., 2006] e pesquisas realizadas utilizando atuadores lineares de acionamento pneumático [Gervini, 2014; Kunz, 2006; Perondi, 2002; Rios, 2009; Sarmanho, 2014], propõe-se um procedimento sistemático para a construção de robôs modulares cartesianos, estatizando os seguintes aspectos: 1) montagem e conexão de robôs cartesianos utilizando módulos mecatrônicos; 2) cálculos da cinemática e dinâmica; 3) síntese de leis de controle, conhecendo a dinâmica do robô; 4) planejamento de trajetória; e, por fim, 5) gerenciamento das unidades baseado no conceito de Multiagente.

4.1 Montagem do robô manipulador

A seguir, a fim de descrever a montagem de um robô manipulador modular, são apresentados os elementos necessários para a construção física de um robô modular por meio de módulos mecatrônicos. Para ilustrar o método proposto, um modelo virtual de um robô modular cartesiano com acionamento pneumático de 3 GDL, é apresentado na Figura 4.1.

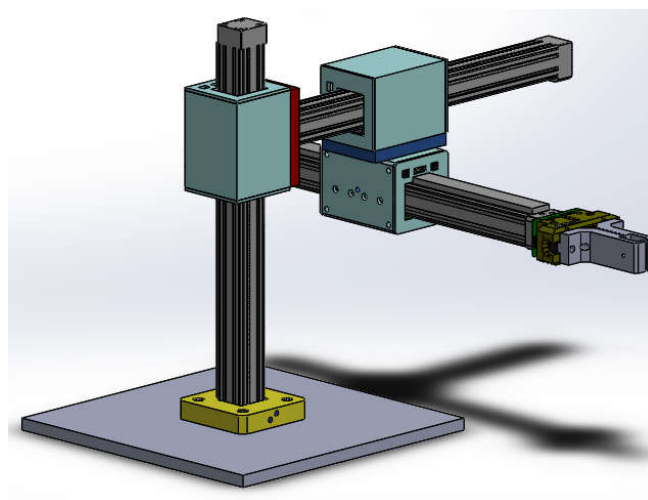


Figura 4.1 – Modelo de robô cartesiano com acionamento pneumático proposto

4.1.1 Elementos para a montagem de um robô modular cartesiano

Inicialmente, antes de descrever o procedimento de montagem, é necessário definir os elementos físicos para a formação de um robô cartesiano de 3 graus de liberdade, foco do presente caso de estudo, que, de acordo com a abordagem, são os seguintes:

- 3 módulos mecatrônicos de acionamento linear;
- 3 módulos auxiliares;
- Parafusos roscados com porca e ruela;
- Base para fixação dos elementos;
- Fonte de ar comprimido (6-10 bar);
- Fonte de energia elétrica (24 Vcc);
- Rede local Ethernet;

Quanto à montagem de um robô modular, assume-se que, além dos módulos mecatrônicos e os módulos auxiliares, existem outros elementos complementares necessários para o funcionamento deste sistema, como, por exemplo, as fontes de energia envolvidas (elétrica e/ou pneumática) e a interface com o usuário. Assim, no escopo do presente trabalho, são analisadas as conexões entre os módulos, a iteração entre os mesmos e os algoritmos de gerenciamentos deste sistema. A Figura 4.2 representa as ligações mecânicas e elétricas entre os elementos que formam parte de um robô manipulador de acionamento pneumático.

O módulo mecatrônico caso de estudo, assim como todo sistema pneumático, precisa minimamente de uma fonte de ar comprimido, filtrado e a uma pressão relativamente constante. Para a montagem um robô utilizando módulos mecatrônicos, ainda existem elementos complementares necessários para o seu funcionamento pleno. O primeiro elo do robô, ligado a uma base fixa, é alimentado com uma fonte de 24Vcc, ainda, todos os módulos devem estar interligados mediante uma rede ethernet, que além dos módulos mecatrônicos, deve existir um outro computador que vai processar alguns dos agentes do sistema Multiagente. Cabe destacar que entre os módulos mecatrônicos existe transmissão de esforços, energia pneumática, energia elétrica e dados, tal como apresentado na Figura 4.2.

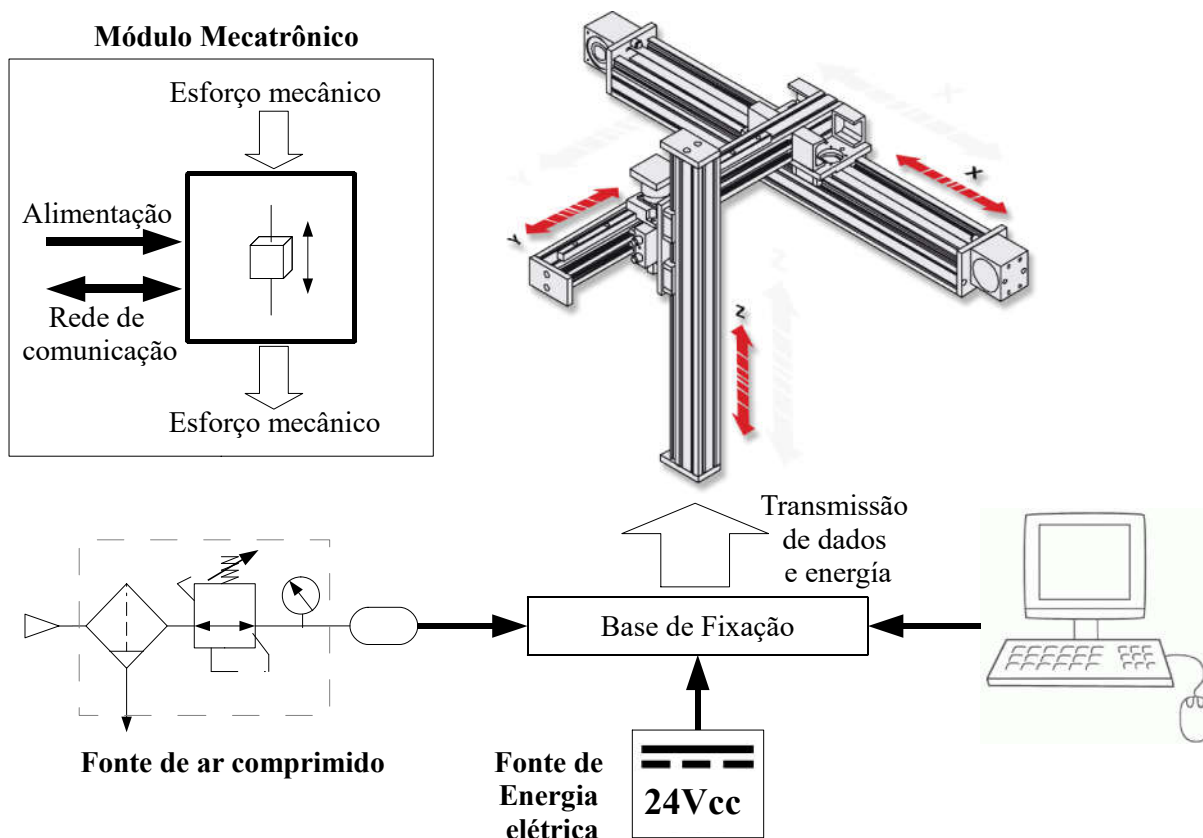


Figura 4.2 – Robô modular e seus periféricos

4.1.2 Descrição dos tipos de conexões entre as unidades

Com relação às ligações mecânicas, os robôs modulares são compostos por três tipos de conexões: 1) conexões mecânicas, as quais fixam elementos físicos para transmissão de esforços e movimentos; 2) conexões de energia, que possibilitam a passagem de energia pneumática entre as juntas; e 3) conexões de transmissão de dados e alimentação, necessárias para a comunicação entre os módulos mecatrônicos e o ambiente externo, e para a alimentação do sistema elétrico de cada unidade mecatrônica.

Um aspecto importante na caracterização das conexões mencionadas consiste no detalhamento do princípio de fornecimento de energia para os módulos mecatrônicos. Nesse sentido, é importante destacar que, embora existam três tipos de conexões, as mesmas podem estar fisicamente dispostas em um único dispositivo físico, possibilitando a transmissão de esforços e movimentos entre dois módulos mecatrônicos, bem como a transferência de energia pneumática. Já, a comunicação de dados e alimentação de energia elétrica dos módulos mecatrônicos são efetuados conectores específicos definidos para esta finalidade. A seguir são detalhadas as ligações mecânicas entre as unidades na formação de uma estrutura robótica,

destacando as conexões mecânicas, os conectores elétricos e as conexões de transferência de energia pneumática.

Conexões mecânicas de fixação base – 1º GDL

Consiste da primeira conexão do sistema modular proposto. É responsável pela sustentação de toda a estrutura do robô. Construtivamente, a conexão mecânica é instalada entre uma base fixa, que deve ser aproximadamente rígida, e o 1º GDL da estrutura modular. A Figura 4.3 apresenta a conexão mecânica entre a base e o primeiro grau de liberdade por meio de parafusos, os quais, através de um módulo auxiliar (módulo passivo específico para a conexão), fixam a base ao primeiro módulo mecatrônico.

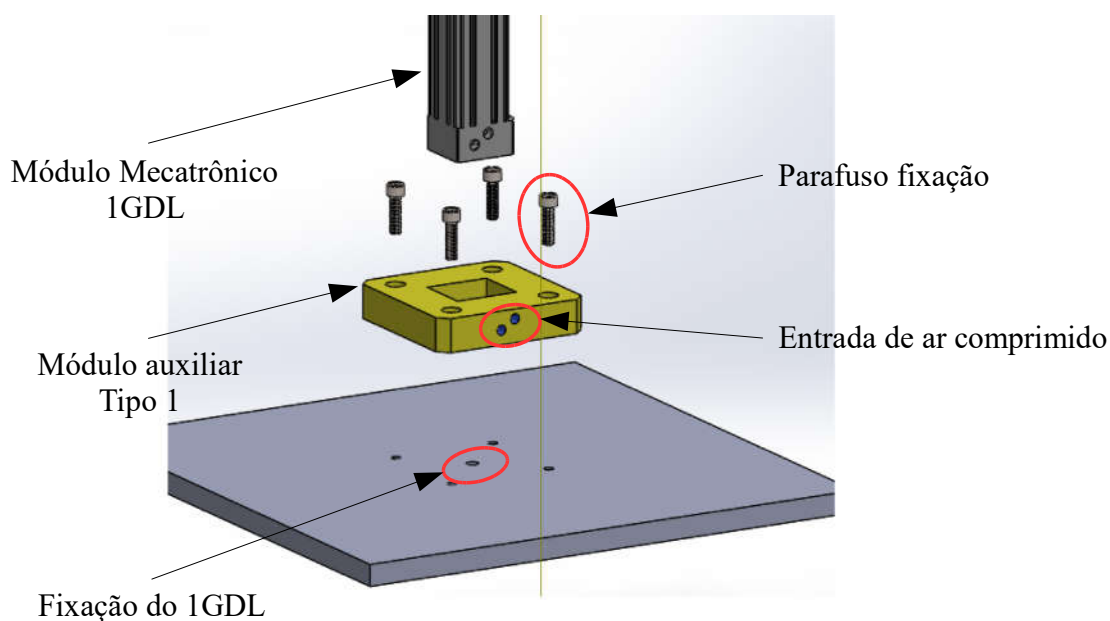


Figura 4.3 – Conexão base com o 1º GDL de acionamento linear

A conexão mecânica, apresentada com detalhe na Figura 4.3, é responsável pela fixação mecânica da estrutura e pela passagem de energia pneumática da base para o primeiro módulo mecatrônico (1º GDL). A base, embora não forme parte do projeto mecânico, deve facilitar a conexão do módulo auxiliar com o 1º GDL através de conectores específicos. Para exemplificar, nesta abordagem utiliza-se conectores na base para a entrada de ar comprimido e pinos para fixação do módulo auxiliar com o módulo mecatrônico.

Conexão de passagem de energia

As conexões de passagem de energia são dispostas juntamente com a conexão mecânica para o acoplamento dos módulos por meio de conectores dispostos para esta finalidade, os quais estão indicados nas figuras 4.3 e 4.5.

Conexões de transmissão de dados e alimentação

Para a transmissão de dados e alimentação de cada módulo mecatrônico são utilizados conectores específicos para atender cada finalidade. Estas conexões são independentes da conexão mecânica. As características deste tipo de conexão foram adaptadas de Larizza et al., 2006, seguindo o pressuposto de que os meios de transmissão de dados e alimentação devem estar separados da conexão mecânica por motivos de segurança de dados, evitando, assim, as interferências eletromagnéticas ocasionadas pelo deslizamentos dos acoplamentos das juntas. A Figura 4.4 ilustra os conectores propostos com as suas especificações.

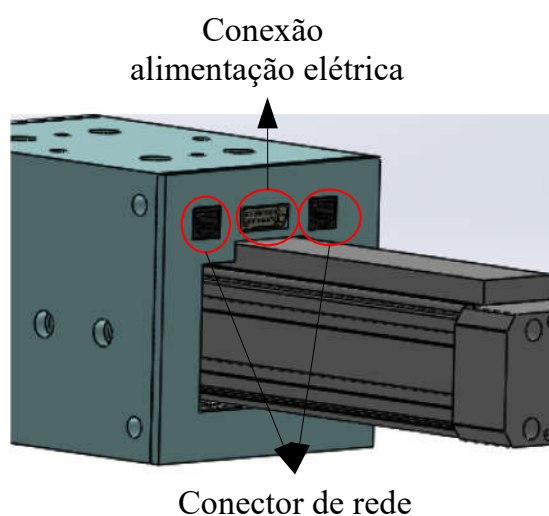


Figura 4.4 – Conexões elétricas e dados

Conexão entre duas unidades mecatrônicas e o efetuator

A fim de estabelecer a conexão entre duas unidades mecatrônicas utiliza-se um modulo auxiliar do tipo 2, conforme apresentado na Figura 4.5.

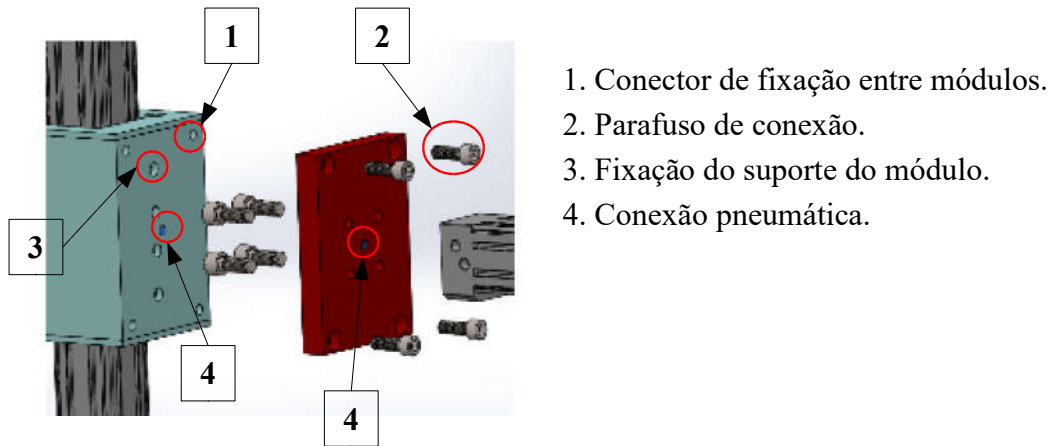


Figura 4.5 – Ligação explodida entre dois módulos mecatrônicos

Também, para a conexão entre um módulo mecatrônico e uma unidade terminal (efetuador), aplica-se um módulo auxiliar do tipo 3, descrito no Capítulo 3, e representado na Figura 4.6.

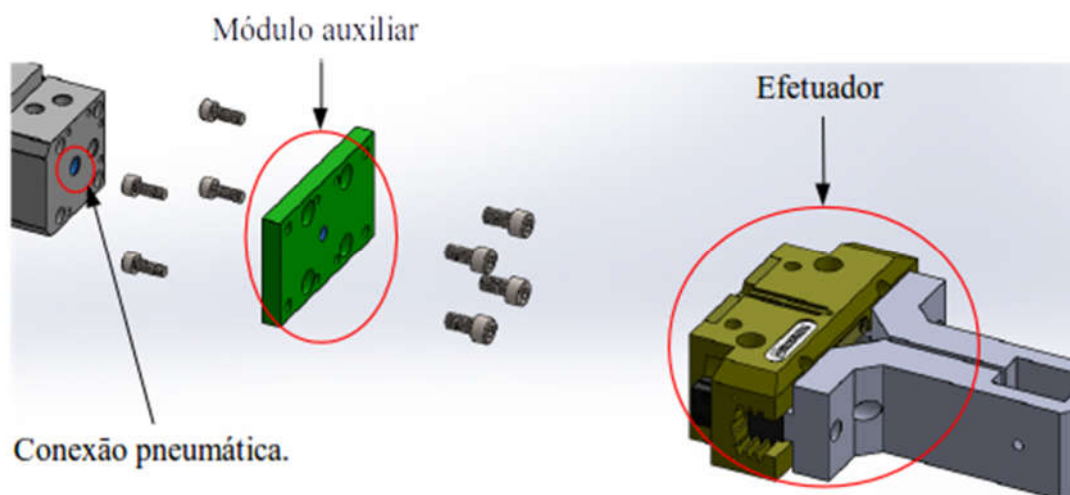


Figura 4.6 – Ligação física entre um efetuator e uma unidade mecatrônica

4.2 Características da cinemática de um robô cartesiano

4.2.1 Cinemática direta

Uma vez realizada a montagem do robô manipulador, da inicialização dos módulos mecatrônicos e da definição do sistema Multiagente, os algoritmos de identificação do sistema, executado pelo agente líder de coalisão, determinam o tipo de estrutura modular formada. Para tanto, a sequência de passos a serem executados até a obtenção da matriz de Denavit-Hartenberg, é apresentada na Figura 4.7.

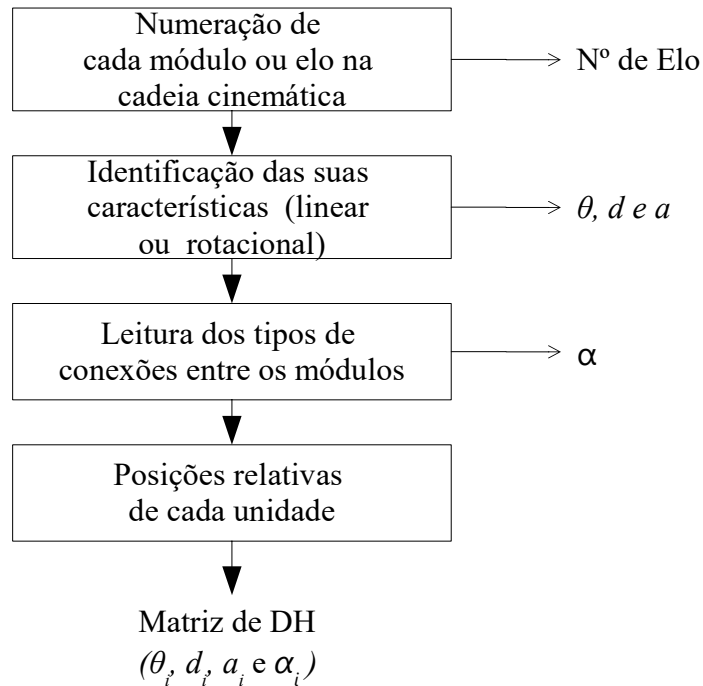


Figura 4.7 – Etapas para a definição da matriz DH

Dentre os parâmetros da matriz de DH, dois destes componentes podem variar em função da movimentação do robô (juntas rotacionais e/ou prismáticas). Por exemplo, se uma junta for rotacional, a variável de junta é o ângulo de junta θ_i , enquanto que, se a junta for prismática, a variável de junta é o deslocamento da junta, ou distância entre os elos, d_i . A Tabela 4.1 resume os conceitos mais importantes sobre os parâmetros cinemáticos relativos à matriz de D-H.

Tabela 4.1- Parâmetros cinemáticos

| Parâmetro | Símbolo | Junta Rotacional | Junta prismática |
|-------------------------|------------|------------------|------------------|
| Ângulo de junta | θ_i | variável | fixo |
| Deslocamento de Junta | d_i | fixo | variável |
| Comprimento do elo | a_i | fixo | fixo |
| Ângulo de torção do elo | α_i | fixo | fixo |

Para efeitos de abordagem genérica, diz-se que o vetor de variáveis de junta é designado por q e dado pela seguinte expressão:

$$\bar{\mathbf{q}} = [q_1 \quad q_2 \quad \dots \quad q_n]^T, \quad q_i = \begin{cases} \theta_i \rightarrow \text{rotacional} \\ d_i \rightarrow \text{prismática} \end{cases}$$

Neste trabalho, considerando a montagem de um robô cartesiano de 3 GDL, de acordo com a cadeia cinemática serial apresentada na Figura 4.1 e o algoritmo apresentado na Figura 4.7, obtém-se, como resposta, a matriz de D-H, apresentada a seguir.

Tabela 4.2 – Matriz de DH para um robô cartesiano de 3º GDL

| Elo | θ_i | d_i | a_i | α_i |
|-----|------------|-------|-------|------------|
| 1 | 90° | d_1 | 0 | 90° |
| 2 | 90° | d_2 | 0 | 90° |
| 3 | 90° | d_3 | 0 | 90° |

De posse das equações 3.3 e 3.5 e conhecendo a matriz D-H, descritas no Capítulo 3, é possível calcular as matrizes de transformação da estrutura robótica proposta. Nas equações 4.1 até a 4.4 são apresentados os procedimentos metodológicos para obtenção desta matriz.

$$T = {}^0A_3 = {}^0A_1 {}^1A_2 {}^2A_3,$$

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} C\theta_i & C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0A_1 = \begin{bmatrix} C\theta_1 & C\alpha_1 S\theta_1 & S\alpha_1 S\theta_1 & a_1 C\theta_1 \\ S\theta_1 & C\alpha_1 C\theta_1 & S\alpha_1 C\theta_1 & a_1 S\theta_1 \\ 0 & S\alpha_1 & C\alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.1)$$

$${}^1A_2 = \begin{bmatrix} C\theta_2 & C\alpha_2 S\theta_2 & S\alpha_2 S\theta_2 & a_2 C\theta_2 \\ S\theta_2 & C\alpha_2 C\theta_2 & S\alpha_2 C\theta_2 & a_2 S\theta_2 \\ 0 & S\alpha_2 & C\alpha_2 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.2)$$

$${}^2A_3 = \begin{bmatrix} C\theta_3 & C\alpha_3 S\theta_3 & S\alpha_3 S\theta_3 & a_3 C\theta_3 \\ S\theta_3 & C\alpha_3 C\theta_3 & S\alpha_3 C\theta_3 & a_3 S\theta_3 \\ 0 & S\alpha_3 & C\alpha_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.3)$$

Assim,

$$T = {}^0A_3 = \begin{bmatrix} 1 & 0 & 0 & d_1 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.4)$$

Os valores limites para os deslocamentos das juntas (módulos mecatrônicos de acionamento linear) são apresentados na Tabela 4.3.

Tabela 4.3 – Valores limites das juntas do robô cartesiano

| Junta | Valores limites |
|-------|-----------------|
| 1 | 0 até 300 [mm] |
| 2 | 0 até 300 [mm] |
| 3 | 0 até 300 [mm] |

4.2.2 Cinemática inversa de um robô cartesiano usando método iterativos

A cinemática inversa, possibilita obter, mediante o conhecimento da posição e orientação desejada do efetuador final, o valor que devem ter os ângulos das juntas do sistema mecânico. O problema da cinemática inversa é fortemente dependente da configuração geométrica do robô, de modo que, para sistemas não redundantes ($n \leq 6$), é recomendável o uso de métodos iterativos para a resolução do sistema de equações, como o método Newton-Raphson.

O método de Newton-Raphson, resolve o problema da cinemática inversa a partir da cinemática direta e é caracterizado pela resolução de sistemas de equações não lineares

mediante métodos numéricos, os quais para um caso geral, possibilitam sua implementação na resolução do problema da cinemática inversa, de acordo com as equações 4.5 e 4.6.

$$P = T Q, \quad (4.5)$$

$$Q = T^{-1}P, \quad (4.6)$$

onde P é o valor da posição no espaço cartesiano, T é a matriz de transformação homogênea e Q é a coordenada de juntas para uma determinada posição. Para um sistema de 3GDL, o qual se refere ao robô cartesiano em estudo, é possível obter um sistema de 3 equações com 3 incógnitas, conforme apresentado na Equação 4.7.

$$\begin{aligned} f_x(q_1, q_2, q_3) & \quad x = 0 \\ f_y(q_1, q_2, q_3) & \quad y = 0. \\ f_z(q_1, q_2, q_3) & \quad z = 0. \end{aligned} \quad (4.7)$$

Ao solucionar a cinemática inversa de robôs manipuladores, os métodos iterativos permitem obter resultados muito próximos aos métodos convencionais de solução cinemática inversa, além de erros usualmente pouco significativos, os quais dependem da rapidez de convergência e dos valores iniciais adotados [Guez e Ahmad, 1988]. Nesse sentido, entende-se que a resolução da cinemática inversa por métodos iterativos pode ser aplicada a um robô cartesiano, bem como para outras configurações de robôs manipuladores de cadeias seriais de até 6 GDL.

Desta forma, propõe-se a aplicação de um método de resolução da cinemática inversa, de acordo com o algoritmo descrito na Figura 4.8. Nesse algoritmo, inicialmente, a partir do conhecimento da cinemática direta e da posição final desejada P no espaço cartesiano, é possível definir as equações diferenciais a serem resolvidas e, caso existam, as restrições e pontos singulares de uma dada configuração. A partir destas informações e a definição de um valor de erro de tolerância, é possível utilizar o método iterativo de Newton-Raphson.

No Anexo B descreve-se a resolução de equações diferenciais mediante métodos iterativos (Newton-Raphson), aplicado para a obtenção da cinemática inversa de um robô manipulador de cadeia serial.

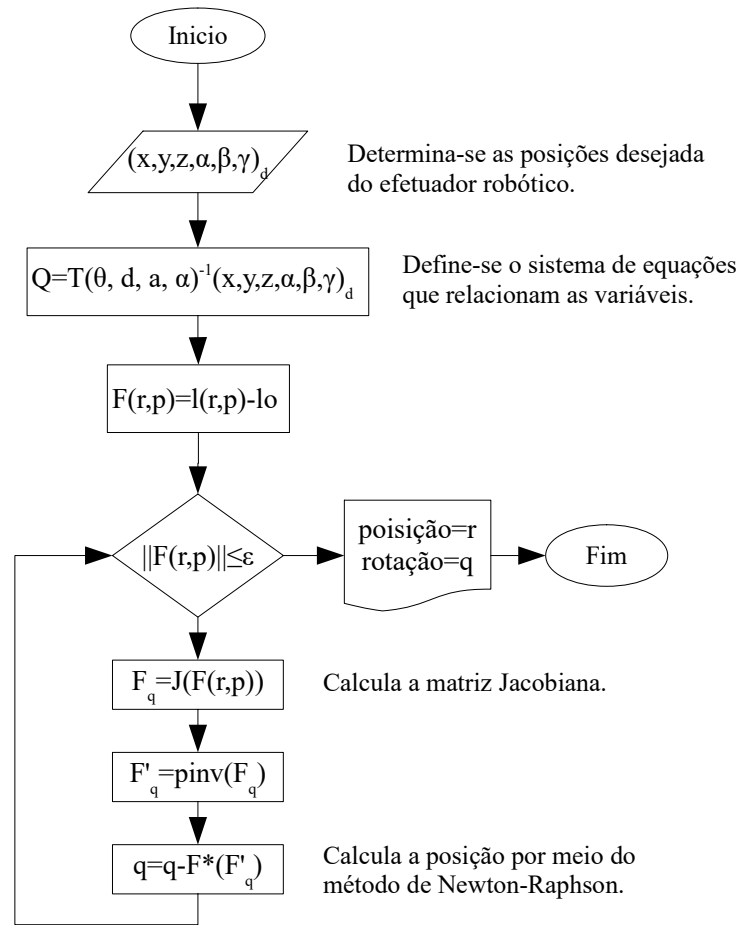


Figura 4.8 – Algoritmo proposto para a execução da cinemática inversa de um robô modular serial

4.3 Dinâmica de um robô rígido

De modo geral, os robôs manipuladores construídos por meio de módulos mecatrônicos tratam-se de estruturas com características próprias, como, por exemplo, a cinemática, a dinâmica, o tipo de acionamento, entre outros, uma vez que são definidos de acordo com os acoplamentos das unidades mecatrônicas, as quais possuem particularidades específicas.

Em relação aos aspectos dinâmicos de estruturas robóticas, existem diversos métodos utilizados na modelagem do movimento de sistemas multicorpos, como as formações apresentadas em Kane, Newton-Euler, Euler-Lagrange, Princípio de D'Alembert ou Hamilton, entre outras, as quais são apresentadas e formuladas em diversos livros-texto na área de robótica [Craig, 2005; Fu et al., 1987; Siciliano et al., 2010; Spong et al., 2005].

Dentre elas, a Formulação Lagrangeana, a qual é objeto do presente trabalho, descreve o modelo dinâmico em termos de energia potencial e cinética por meio de um sistema de coordenadas generalizadas. Esse método, em conjunto com a notação de Denavit-Hartenberg,

pode ser utilizado nos cálculos da dinâmica do sistema, bem como nos cálculos de torques das juntas, além de não apresentarem custo computacional elevado, apesar de sua implementação necessita de um grande número de operações matemáticas, devido às derivações numéricas necessárias [Fu et al., 1987].

4.3.1 Modelo dinâmico de um robô com elos rígidos

De acordo com Slotine, 1988, um modelo dinâmico de um robô serial com elos rígidos, considerando a parcela dinâmica dos atuadores presentes nas juntas, pode ser expresso por equações que relacionam os torques e forças gerados pelos atuadores e o movimento por eles causado nos elos da estrutura.

Nesta perspectiva, considerando um robô com n graus de liberdade que opera na ausência de forças externas aplicadas a seu efetuador e com atrito desprezível nas juntas, assume-se que o modelo matemático da dinâmica no espaço das juntas, que é derivada da aplicação da Formulação Lagrangeana, pode ser expresso em uma forma matricial compacta, conforme a Equação 4.8.

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (4.8)$$

onde q é o vetor de coordenadas generalizadas das juntas, $H(q)$ é a matriz de inércia simétrica definida positiva e, em geral, dependente da configuração do robô, $C(q, \dot{q})$ é a matriz que representa os efeitos centrífugos e de Coriolis, $G(q)$ é o vetor que representa o momento gerado em cada eixo de junta do manipulador devido a presença da gravidade, e τ é vetor de torques ou forças de atuação nas juntas.

4.3.2 Controle Independente por Junta

O controle independente por junta consiste em considerar cada junta independente das demais para efeitos de controle. Dessa forma, projeta-se um controlador para cada junta, ignorando os efeitos de acoplamento entre uma junta com as demais. Já para o controle do sistema considerado acoplado, é necessário um modelo detalhado do sistema dinâmico que é utilizado na síntese do controlador a prova de estabilidade em malha fechada, o que incrementa a complexidade do algoritmo de controle [Oliveira e Lages, 2006].

A Figura 4.9 mostra um diagrama de blocos de um controle independente por junta. Para analisar o controle independente por junta utilizam-se usualmente um modelo do sistema de uma junta.

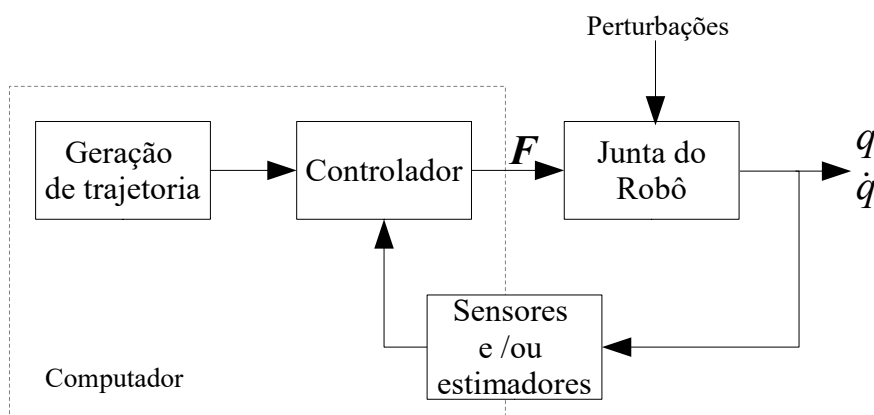


Figura 4.9 – Diagrama de blocos genérico de um sistema de controle independente por junta de um robô manipulador [adaptado de Oliveira e Lages, 2006].

Algoritmo de controle aplicado a robôs manipuladores baseados no Algoritmo de Slotine e Li

Em Bobrow e McDonell, 1998, é apresentada uma extensão da conhecida Lei de Slotine e Li para aplicação do controle em manipuladores robóticos acionados eletricamente, adaptando estes algoritmos para o controle em manipuladores acionados pneumaticamente. Esta lei é também conhecida como lei de Torque Calculado (TC). Uma adaptação da lei de controle de Slotine, 1988, para robôs manipuladores pneumáticos com compensação de atrito foi proposta por Sarmanho, 2014, para o controle global de um robô manipulador de 5 GDL, e cujas formulações são consideradas neste trabalho. Estratégias semelhantes de controle foram aplicadas em sistemas pneumáticos de um grau de liberdade, tais como os apresentados por Perondi, 2002; Sobczyk, 2009.

No presente trabalho propõe utilizar uma variante do algoritmo de Torque Calculado (TC), no qual permite o controle de sistemas pneumáticos a partir de estimativas das forças aplicadas nas juntas e do conhecimento prévio do modelo dinâmico do sistema, o qual, como já mencionado, é obtido por meio das informações fornecidas pelas unidades mecatrônicas.

Slotine, 1988, define que o vetor q é composto por variáveis que representam as posições das juntas do manipulador e o vetor q_d as posições da trajetória desejada ao longo do

tempo para as juntas. Assim, os termos do vetor de erros de seguimento de posição e suas derivadas podem ser expressos por meio das equações 4.9, 4.10 e 4.11:

$$e = q - q_d, \quad (4.9)$$

$$\dot{e} = \dot{q} - \dot{q}_d, \quad (4.10)$$

$$\ddot{e} = \ddot{q} - \ddot{q}_d. \quad (4.11)$$

De acordo com Slotine, 1988, o Torque Calculado (TC) pode ser expresso através de uma equação que permite obter os valores dos torques necessários para que cada grau de liberdade do manipulador siga adequadamente as respectivas trajetórias desejadas, tal como apresentado na Equação 4.12:

$$\tau_s = H(q)\ddot{v} + C(q, \dot{q})\dot{v} + G(q) - K_d(\dot{q} - \dot{v}), \quad (4.12)$$

onde K_d é uma matriz simétrica positiva definida, os vetores v , \dot{v} e \ddot{v} são expressos através das equações 4.13, 4.14 e 4.15, que consiste de vetores relativos aos erros ponderados entre as velocidades desejadas. Os erros de posição podem ser definidos como:

$$v = q_d - \lambda e, \quad (4.13)$$

$$\dot{v} = \dot{q}_d - \lambda \dot{e}, \quad (4.14)$$

$$\ddot{v} = \ddot{q}_d - \lambda \ddot{e}, \quad (4.15)$$

onde λ é definida como uma matriz simétrica positiva.

A primeira derivada no tempo da Equação 4.12 pode ser escrita como a Equação 4.16.

$$\tau_s = H(q)\ddot{v} + \dot{H}(q)\dot{v} + C(q, \dot{q})\dot{v} + G(q) - K_d(\dot{q} - \dot{v}). \quad (4.16)$$

Já, as matrizes $H(q)$, $C(q, \dot{q})$ e $G(q)$ definem as características dinâmicas de uma determinada estrutura.

Os erros de torques e forças nas juntas ($\tilde{\tau}$) e sua derivada $\dot{\tilde{\tau}}$, desconsiderando a presença de forças de atrito nos sistemas mecânicos, podem ser definidos como apresentadas as equações 4.17 e 4.18:

$$\tilde{\tau} = \tau - \tau_s, \quad (4.17)$$

$$\tilde{\tau} = \tau - \tau_s. \quad (4.18)$$

Assim, baseada na equação que rege a dinâmica do manipulador e a definição do erro de força/torque, obtém-se a Equação 4.19:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tilde{\tau} + \tau_s. \quad (4.19)$$

Substituindo a lei de controle de Slotine, 1988, (Equação 4.8) na Equação 4.19, obtém-se:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tilde{\tau} + H(q)\dot{v} + C(q, \dot{q})\dot{v} + G(q) - K_d(\dot{q} - v). \quad (4.20)$$

A Equação 4.20 pode ser escrita como:

$$H(q)(\ddot{q} - \dot{v}) + C(q, \dot{q})(\dot{q} - v) + K_d(\dot{q} - v) - \tilde{\tau} = 0. \quad (4.21)$$

Assim como em Slotine, 1988, o vetor dos erros de seguimento de posição e velocidade e suas derivadas podem ser escritos de acordo com as equações 4.15 e 4.16:

$$s = e + \lambda e = \dot{q} - v, \quad (4.22)$$

$$\dot{s} = \dot{e} - \lambda \dot{e} = \ddot{q} - \dot{v} \quad (4.23)$$

Agrupando as equações 4.21, 4.22 e 4.23, obtém-se a Equação 4.24.

$$H(q)s + C(q, \dot{q})s + K_d s - \tilde{\tau} = 0. \quad (4.24)$$

Para a análise de estabilidade, propõe-se a seguinte candidata a função de Lyapunov (Equação 4.25):

$$V = \frac{1}{2} [s^T H(q)s + \tilde{\tau}^T K_1^{-1} \tilde{\tau}], \quad (4.25)$$

cuja primeira derivada no tempo é:

$$V = \frac{1}{2} s^T H(q) s + s^T H(q) s + \tilde{\tau}^T K_1^{-1} \tilde{\tau}. \quad (4.26)$$

Substituindo a Equação 4.24 na Equação 4.26 e trabalhando algebricamente, tem-se a Equação 4.27:

$$V = s^T K_d s + s^T \tilde{\tau} + \frac{1}{2} s^T [H(q) \quad 2C(q, q)] s + \tilde{\tau}^T K_1^{-1} \tilde{\tau}. \quad (4.27)$$

O sistema $H(q) \quad 2C(q, q)$ é uma matriz anti-simétrica, assim, $\frac{1}{2} s^T [H(q) \quad 2C(q, q)] s = 0$, o que resulta na Equação 4.28:

$$V = s^T K_d s + s^T \tilde{\tau} + \tilde{\tau}^T K_1^{-1} \tilde{\tau}, \quad (4.28)$$

Dependente das forças e torques e da derivada primeira aplicados nas juntas. Assim, buscando seguir a trajetória desejada, supondo que as matrizes de forças e torques são conhecidas, são substituídas na Equação 4.28, as equações 4.17 e 4.18 e se obtém a Equação 4.29:

$$V = s^T K_d s + s^T (\tau \quad \tau_s) + \tilde{\tau}^T K_1^{-1} (\tau \quad \tau_s), \quad (4.29)$$

a qual, expandida, resulta na Equação 4.30.

$$V = s^T K_d s + s^T \tau \quad s^T \tau_s + \tilde{\tau}^T K_1^{-1} \tau \quad \tilde{\tau}^T K_1^{-1} \tau_s \leq 0. \quad (4.30)$$

Lembrando que os torques e/ou forças são fornecidos pelos elos envolvidos na estrutura do robô (módulos mecatrônicos).

É necessário ainda provar que $V \rightarrow 0$ quando $t \rightarrow \infty$ e, desta forma, que os vetores de erros $s \rightarrow 0$ e $\tau \rightarrow 0$ quando $t \rightarrow \infty$. Isto pode ser feito por meio do Teorema de Barbalat, segundo o qual, se $V \geq 0$ e $\dot{V} \leq 0$, $V \rightarrow 0$ se V é uniformemente contínua, ou seja, se V é limitada. Para tanto, deriva-se a Equação 4.30, resultando em:

$$V = s^T K_d s + s^T \tau - s^T \tau_s + \tilde{\tau}^T K_1^{-1} \tau - \tilde{\tau}^T K_1^{-1} \tau_s. \quad (4.31)$$

Portanto, para mostrar que V é limitada é necessário mostrar que s , \dot{s} e $\tilde{\tau}$ são limitados. Como $V \geq 0$ e $\dot{V} \leq 0$, V se mantém limitada e, da Equação 4.17, conclui-se que os vetores s e \dot{s} devem ser limitados; e, como $s = e + \lambda e$, resulta que, quando $\lambda > 0$, essa equação representa um sistema do tipo BIBO (*Bonde Input Bonde Output*) linear de primeira ordem, com entrada definida como s e saída como e . Logo, se $s \rightarrow 0$ então $e \rightarrow 0$ e $\dot{e} \rightarrow 0$. Como consequência, uma vez que os valores desejados para as trajetórias de juntas são, por hipótese, limitados, então q e \dot{q} são também limitados. Como a matriz de inércia $H(q)$ é inversível, a Equação 4.22 pode ser reescrita como:

$$\ddot{s} = H^{-1}(q) [-C(q, \dot{q})\dot{s} - K_d \dot{s} + \tilde{\tau}]. \quad (4.32)$$

Assim, pode-se concluir que s é limitada pelos valores de s e $\tilde{\tau}$. Também o valor de V é limitado, e, de acordo com o Lema de Barbalat, $V \rightarrow 0$ quando $t \rightarrow \infty$, e, como consequência, $s \rightarrow 0$ e $\dot{s} \rightarrow 0$ quando $t \rightarrow \infty$. Quando $s \rightarrow 0$, os erros de seguimento de posição e velocidade também tendem a zero e, quando $\tilde{\tau} \rightarrow 0$, a força aplicada tende ao valor soma da força da Lei de Controle de Slotine e Li, ou seja, $\tau \rightarrow \tau_s$, completando a prova.

Cabe destacar que o torque aplicado a cada junta não pode ser medido diretamente, onde a força que realiza cada atuador pode ser calculada através da diferença de pressão entre as câmaras (p). Desta forma, para um atuador simétrico (mesma área efetiva dos dois lados do êmbolo), a força proporcional ao torque, pode ser definido como $F_p = A p = A(p_a - p_b)$.

Bobrow e McDonell, 1998, mostra que, em sistemas com acionamento pneumático, a combinação linear dos erros de seguimento de posição e velocidade tende a zero quando é aplicada a Lei de Slotine e Li com parâmetros conhecidos.

4.4 Planejamento de trajetórias aplicadas a robôs modulares

Para que um robô modular possa executar suas tarefas, é necessário conhecer previamente as características da estrutura modular formada, de forma que seja possível planejar suas ações. Assim, uma vez que o Sistema Multiagente definiu as características de manipulação (cinemática, volume de trabalho, carga máxima, etc.), o sistema de gerenciamento, por meio da execução de agentes com funções específicas, realiza, dentre as tarefas

mencionadas no Capítulo 3, o processamento dos algoritmos de planejamento de trajetória. Essas trajetórias são definidas para cada junta e devem, para a adequada operação do robô obedecer a critérios pré-estabelecidos que delimitam as faixas de operação do sistema, como velocidade, acelerações e *jerks* máximos.

Dentre diversas possibilidades de algoritmos de geração de trajetória, optou-se por se tratar de um estudo preliminar, por um esquema baseado na utilização de curvas geradas por meio de interpolação por *splines* de 3º grau. Dentre diversas técnicas podem ser utilizadas na resolução dos problemas de interpolação por *splines* de 3º grau, como apresentado na Equação 3.33:

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i. \quad (4.33)$$

Autores como Chapra e Canale, 2008; Ruggiero e Lopes, 1996, entre outros, mostram que, para obtenção de polinômios de terceiro grau entre cada intervalo de nós, é necessários atender alguns requisitos.

A fim de que os pontos sejam interpolados, existem n intervalos e, por consequência, incógnitas para serem obtidas. As condições impostas para a *spline* são:

- a) os valores das funções devem ser iguais nos nós internos ($2n-2$ condições);
- b) as funções do primeiro e do último segmento devem passar, respectivamente, pelos pontos inicial e final (2 condições);
- c) as primeiras derivadas nos nós internos devem ser iguais ($n-1$ condições);
- d) as derivadas segundas nos nós internos devem ser iguais ($n-1$ condições); e
- e) a segunda derivada no nó inicial e no final devem ser nulas (2 condições).

A última condição, aplicada aos nós extremos da função, caracteriza a chamada *spline* natural, de forma que, se necessário, pode-se fazer com que a segunda derivada tenha valor diferente de zero nas extremidades da função.

4.5 Proposta do sistema Multiagentes em módulos mecatrônicos

Para elucidar os conceitos relativos à arquitetura do Sistema Multiagente proposto, primeiramente, é necessário descrever a plataforma (software) de desenvolvimento de agentes de forma genérica. Neste trabalho, utiliza-se como referência a plataforma JADE (*Java Agent*

Development), a qual é base para a criação da plataforma de desenvolvimento de agentes mecatrônicos.

4.5.1 Java Agent DEvelopment – JADE

JADE é um conjunto de bibliotecas de classes que podem ser executadas em um Ambiente de Desenvolvimento Integrado (IDEs), como o Netbeans [NetBeans, 2016]. Estas bibliotecas podem ser utilizadas na implementação dos agentes, visto que permitem a comunicação mediante a utilização do protocolo FIPA (descrito no Capítulo 3). Ademais, o JADE facilita a programação Multiagente na forma de um pacote de linguagem Java.

Pode-se compreender a plataforma JADE como um conjunto de *containers* que facilitam os serviços necessários para a execução e comunicação de agentes. Um *container* é uma máquina virtual Java independente, podendo ser interpretado como um nó em uma rede.

A seguir, com a finalidade de descrever o princípio de operação e comunicação das entidades ou agentes, são apresentados os conceitos e principais comportamentos dos agentes no ambiente JADE, além dos principais tipos de operações, e da forma de destes sistemas.

Ciclo de vida de um agente

Dentro de um *container* JADE, um agente é modelado através de uma classe Java, a qual implementa um *thread* Java (processamento paralelo) que executa comportamentos continuamente. Um *thread* é um processo paralelo, é um pequeno programa que trabalha como um subsistema, sendo uma forma de um processo se autodividir em duas ou mais tarefas.

Os agentes FIPA existentes em uma plataforma de agentes possuem um ciclo de vida controlado pelo um *thread* do sistema Multiagente. O ciclo de vida de um agente FIPA possui as seguintes características [FIPA, 2002]:

- Limitado a uma plataforma de agentes: Um agente é fisicamente gerenciado dentro de uma plataforma de agentes e, portanto, o seu ciclo de vida é sempre limitado a essa plataforma de agentes;
- Independente de aplicações: O modelo de ciclo de vida é independente de qualquer aplicação, definindo somente os estados e as transições de estados ao longo da vida de um agente;

- Orientado à instância: O agente descrito no modelo de ciclo de vida é uma instância de um agente, o que significa que um agente possui um nome único e é executado de forma independente;
- Único: Cada agente possui apenas um estado de ciclo de vida em qualquer momento, e dentro de somente uma plataforma de agentes.

A Figura 4.10 ilustra o ciclo de vida de um agente FIPA. Os possíveis estados de um agente são descritos a seguir [FIPA, 2002]:

- Iniciado: o agente já foi criado, porém ainda não é capaz de atuar ou interagir com os outros agentes.
- Ativo: o agente registrou-se com o AMS recebendo um AID válido, tornando-se capaz de interagir com os demais agentes. No estado ativo o agente está apto a receber e enviar mensagens a qualquer outro agente.
- Suspenso: o agente fica temporariamente incapaz de executar suas atividades. Esta transição pode ser causada pelo AMS ou pelo próprio agente.

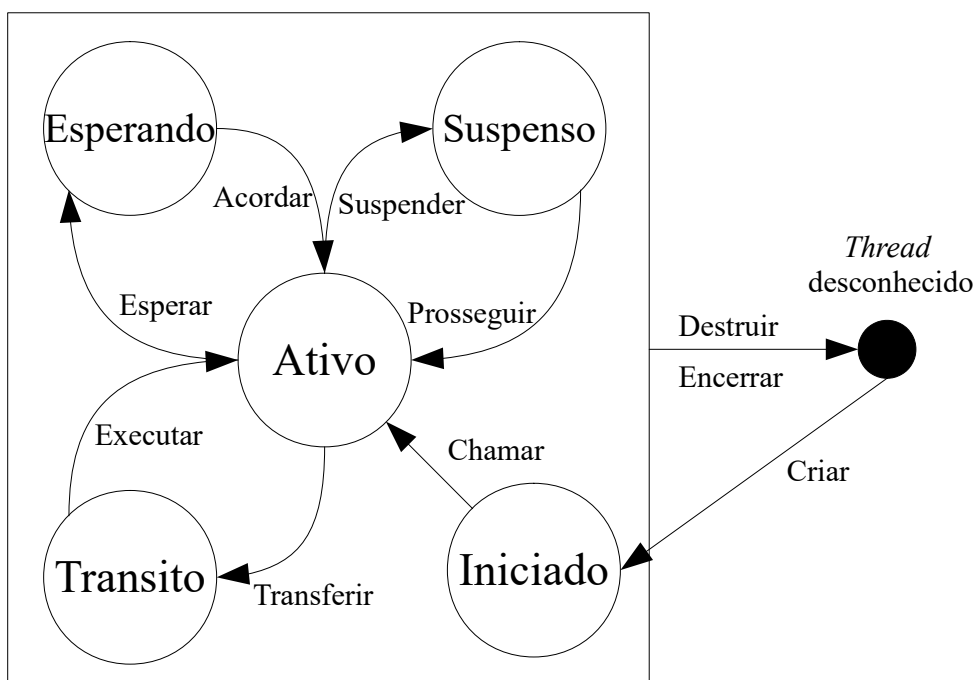


Figura 4.10 – Ciclo de vida de um agente FIPA [FIPA, 2002]

- Esperando: o próprio agente decide ficar temporariamente inativo enquanto espera que um evento externo ocorra.

- Trânsito: é um estado exclusivo de agentes móveis. O agente móvel é capaz de descocar-se para outro host e continuar sua execução em uma máquina diferente. Nesta abordagem não serão tratados estes casos.

Para que o agente execute alguma tarefa, o programador deve sobrepor ao menos dois métodos da classe Agent: *setup()* e *takeDown()*. O primeiro é chamado quando da iniciação do agente, o último é chamado quando da finalização do agente.

Durante o tempo de vida do agente, o JADE irá executar os códigos relativos aos comportamentos associados àquele agente. Um comportamento é um código de execução compartilhada dentro do agente, modelada através da classe *Behaviour*. Um agente pode ter vários comportamentos que são executados na forma de tempo compartilhado por cooperação. Os comportamentos são normalmente adicionados ao agente no método *setup()*, mas podem ser adicionados a partir de qualquer rotina do agente, inclusive a partir de outros comportamentos. Para isso, o JADE conta com a variável *myAgent* associada a cada comportamento, a qual aponta para o agente hospedeiro daquele comportamento.

A ideia por detrás do modelo de comportamentos em classes é ter uma ferramenta capaz de definir o aspecto funcional e dinâmico do agente. Em um comportamento é possível definir-se uma função específica, como a captura de dados externos, análise da situação corrente, definição da ação no meio ou a ação própria. É importante destacar que o comportamento global (conceitual) será o resultado das ações de todos os comportamentos JADE *Behaviours* do agente JADE [JADE Board, 2016].

O ciclo de execução de um agente é mostrado na Figura 4.11, na qual pode-se verificar que os comportamentos são executados continuamente, pela execução do método *action()* do comportamento tomado da fila de comportamentos ativos. Quando um comportamento é bloqueado, ele sai da fila de ativos e vai para a fila de bloqueados.

A política de varredura das filas no JADE é cíclica. Assim que um comportamento é retirado do início da fila, é executado e colocado no final da fila, fazendo com que o JADE execute todos os comportamentos ativos ciclicamente. A adição de um comportamento acontece sempre ao final da fila circular.

O término da execução de um comportamento é definido pelo retorno do método *done()*. Enquanto retorna estágio *false*, o comportamento é mantido em uma das listas de comportamentos, e quando retorna *true*, o comportamento é removido das listas de comportamentos do agente.

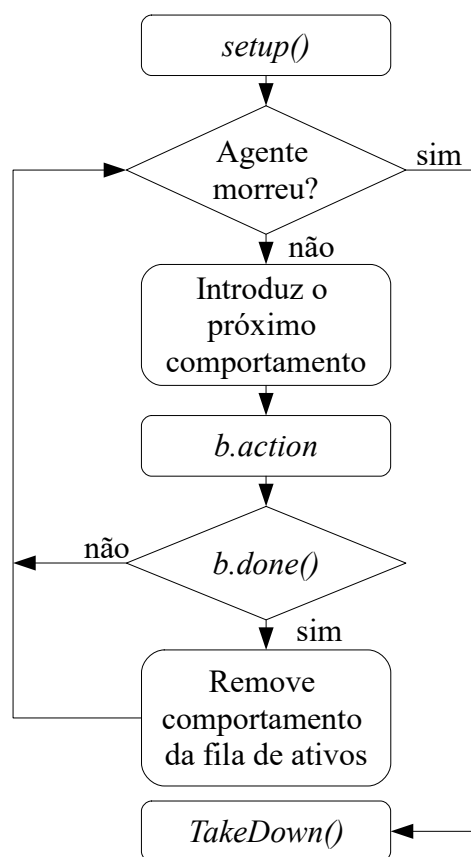


Figura 4.11- Execução de um agente JADE.

Um comportamento pode ser colocado em estado de bloqueio explicitamente pela chamada do método *block()* ou implicitamente através da entrada em espera por comunicação (por exemplo *myAgent.receive()*). Um comportamento é colocado em estado de ativo quando for o primeiro da fila de bloqueados, mas está pronto, isto é, não está à espera de nenhum evento externo (como por exemplo, eventos de comunicação). Dessa forma, um agente sempre “acorda” ao final de um ciclo de execução de todos os comportamentos, quando recebe uma mensagem de comunicação. Uma situação potencialmente perigosa é quando todos os comportamentos são bloqueados e não há uma mensagem de comunicação. Neste caso, o agente aparenta “congelar”. A plataforma JADE define alguns comportamentos padrões que podem ser implantados usando rotinas disponibilizadas, que são as seguintes:

- **SimpleBehaviour:** é a implementação da classe abstrata *Behaviour*. O programador deve sobrepor tanto o método *action()* quando o *done()*. É um comportamento básico propriamente falando. O comportamento *SimpleBehaviour* é responsável por executar uma determinada tarefa ciclicamente até que esta termine. Um exemplo é pedido que se faça um *SimpleBehaviour* que corra quatro vezes até terminar (contador ou incrementador).

- ***CyclicBehaviour***: implementa um *SimpleBehaviour* com o método *done()* sempre retornando *false*. Dessa forma, o comportamento nunca termina (mas pode ser bloqueado implícita ou explicitamente). Este comportamento pode ser executado se durante a execução de um comportamento este demorar muito tempo a processar informação ou a realizar uma determinada tarefa, os restantes *behaviours* ficam bloqueados e o agente também “congelado”. Como tal os ciclos codificados dentro dos comportamentos devem ser implementados recorrendo a *Cyclic Behaviours*, que em cada chamada, correm uma iteração, ciclicamente.
- ***OneShotBehaviour***: implementa um *SimpleBehaviour* com o método *done()* sempre retornando *true*, isto é, o método *action()* do comportamento é executado apenas uma vez. O *OneShotBehaviour* quando é adicionado é responsável por fazer uma tarefa uma única vez. Na prática é como se chamasse um *SimpleBehaviour* mas em que o método *done()* retorna *true*, terminando assim a execução logo na primeira chamada.
- ***WakerBehaviour***: implementa um comportamento que executa uma ação (*onWake()*) após uma quantidade de tempo definida na criação. O *WakerBehaviour* tem um comportamento idêntico ao *OneShotBehaviour*, com a diferença que quando este é chamado recebe como parâmetro um valor do tipo *long* que indica o tempo (em milissegundos) de espera até que este comportamento seja disparado. Ou seja, se um comportamento deste gênero for lançado com valor 1000 no parâmetro no construtor deste *behaviour* este ficará “adormecido” durante 1000 milissegundos até ser disparado e correr uma vez
- ***TickerBehaviour***: implanta um comportamento que executa uma ação (*onTick()*) periodicamente, isto é, sempre que um timer, definido na criação, estoura. O timer pode ser parado ou reiniciado e o período pode ser variado durante a execução da ação. O *TickerBehaviour* permite correr uma tarefa periodicamente, ou seja, à semelhança ao que acontece com o *CyclicBehaviour*, este arranca e corre ciclicamente desde a instanciação até que o agente deixa a plataforma. A diferença deste *behaviour* para o *CyclicBehaviour* é que existe uma janela de tempo entre execuções.
- ***SequentialBehaviour***: executa subcomportamentos de modo sequencial, isto é, a cada ciclo o próximo subcomportamento quando o atual termina (seu método *done()* retorna *false*). Termina quando o último subcomportamento termina.
- ***FSMBehaviour***: executa subcomportamentos como uma máquina de estados em que os eventos são os valores de retorno do método *onEnd()* e cada subcomportamento é

um estado da máquina. É a base para os comportamentos que implementam os protocolos FIPA.

4.6 Considerações finais

Após a montagem e alimentação do robô modular na configuração cinemática desejada, é executada a etapa de negociação e formação do sistema Multiagente. De acordo com a abordagem proposta, este estágio, o sistema Multiagente que realiza o gerenciamento do manipulador é responsável pelo cálculo do modelo matemático do robô em questão com base nos dados fornecidos por cada agente do sistema, como, por exemplo, o número da junta, o tipo de acionamento, o posicionamento e a orientação do robô. No presente estudo são apresentados procedimentos para obtenção de modelos necessários para a implantação automática do algoritmo de controle por torque calculado e planejamento de trajetória. Evidentemente, outras estratégias de controle (por exemplo, controle por modos deslizantes, por *backstepping*, adaptativos, com compensação de atrito, dentre outras estratégias) deverão ser objeto de estudos futuros. É importante destacar que técnicas robustas de controle deverão ter atenção especial, seja esta a grande variabilidade de estruturas que poderão vir a ser concebidas, o que pode dificultar o conhecimento prévio preciso dos parâmetros físicos geométricos de robôs montados.

Os algoritmos da cinemática direta e inversa para cálculo do robô modular foram propostos para aplicação em cadeias cinemáticas seriais. Os resultados das simulações realizadas são considerados satisfatórios tanto para estruturas cinemáticas cartesianas, como cilíndricas com 3 GDL. Ainda que este trabalho não apresente os resultados relativos a outras configurações cinemáticas ou aos casos com maior quantidade de juntas, entende-se que a cinemática direta não apresenta inconvenientes para aplicações nestes outros casos, pois, como já mencionado, a identificação da matriz de DH é realizada pela chamada de uma função genérica executada pelo agente líder de coalisão (comportamento simples ou *simple behaviour*) da plataforma Multiagente. No entanto, a resolução da cinemática inversa para sistemas com mais de 3 GDL pode resultar em um sistema de equações não lineares que, resolvendo por métodos numéricos, podem levar a resultados imprecisos. Isto deve-se ao fato que os métodos iterativos de Newton-Raphson, quando aplicados em sistemas de equações de maior complexidade, não convergir para valores fisicamente consistentes.

Na implementação do robô modular, além dos seus aspectos construtivos e operacionais, é importante dispor de algoritmos de planejamento de trajetória aplicáveis a

diferentes tipos de plataformas (robôs cartesianos, cilíndricos, SCARA, etc.). Como resultado da pesquisa realizada, entende-se que não há necessidade de alterações ou adaptações dos algoritmos pesquisados na implantação física dos sistemas modulares, visto que, independentemente do tipo de configuração do robô, o planejamento de trajetória depende exclusivamente do conhecimento da cinemática do robô montado, o qual, como visto no trabalho, pode ser adequadamente obtido aplicando os procedimentos propostos.

O sistema de controle é, na implementação do sistema físico, uma das funcionalidades com maior desafio na implantação, pois são suscetíveis a apresentar comportamento instável em função da possível variação de parâmetros associados ao comportamento não-linear do sistema físico. Ainda, devido ao fato de os algoritmos de controle implantados não considera explicitamente alguns termos tais como o atrito, os quais podem levar ao sistema a cometer erros no seguimento de trajetória. Para tanto, estudos estratégicos para sintonização dos ganhos e de verificação da estabilidade dos controladores, são importantes e deverão ser objeto de futuros trabalhos.

5 ESTUDO DE CASO E PROCEDIMENTO SISTEMÁTICO PARA DEFINIÇÃO DE ROBÔS MODULARES

Levando-se em conta os conceitos apresentados nos capítulos 3 e 4, o presente capítulo objetiva descrever a sequência de passos para construção de um robô modular funcional. Nesta abordagem, a fim de descrever o método proposto, apresenta-se como estudo de caso, um robô cartesiano elaborado por meio módulos mecatrônicos com acionamento pneumático. A partir desse estudo de caso, foi possível propor um procedimento para elaboração de robôs modulares, o qual pode ser sintetizado por meio dos seguintes passos: (i) definição da sequência de montagem de um robô modular a partir de módulos mecatrônicos; (ii) geração de trajetória para cada junta, de acordo com os conceitos apresentados na Seção 4.3; (iii) seguimento de trajetória (controle de posição) por meio de controladores baseados em modelos matemáticos do robô modular; e, finalmente, (iv) definição da arquitetura do sistema Multiagente a ser executado no robô modular, bem como da interface para a verificação dos módulos ativos do sistema.

A seguir é apresentado o desenvolvimento do robô cartesiano (estudo de caso) que embasou o desenvolvimento do procedimento sistemático.

5.1 Estudo de caso: robô modular cartesiano

Um estudo preliminar básico indica que para a formação de um manipulador cartesiano de 3 GDL é necessário dispor-se dos seguintes elementos:

- 3 módulos mecatrônicos de acionamento lineares;
- 4 módulos auxiliares:
 - 1 módulo auxiliar tipo 1 (acoplamento entre a base e o 1º GDL);
 - 2 módulos auxiliares tipo 2 (acoplamento entre os 1º, 2º e 3º GDL);
 - 1 módulo auxiliar tipo 3 (acoplamento entre o 3º GDL e o efetuador);
- 1 fonte de alimentação para os processadores dos módulos mecatrônicos;
- 1 fonte de energia para o acionamento (ar comprimido);
- 1 rede local ethernet;
- 1 computador conectado à rede local.

A definição destes componentes foi realizada empiricamente, porém, é relativamente simples se estabelecer alguma regra geral para definição dos elementos necessários para diversas configurações possíveis de robôs. Não se considera, por outro lado, que essa seja a

melhor opção, pois poderia “engessar” em demasia o procedimento de definição dos componentes, dificultando a definição quando peculiaridades deferentes às aplicações devem ser consideradas.

Para embasar a definição dos componentes, deve-se levar em consideração que, conforme mencionado na Seção 4.2, existem alguns tipos de conexões predefinidas que, dependendo do tipo de acionamento das juntas a serem utilizadas (prismáticas ou rotacionais), permitem dois tipos de acoplamentos, os quais possibilitam a formação de manipuladores com diferentes geometrias, como, por exemplo, o robô cartesiano, o robô SCARA, o robô cilíndrico, entre outros.

Para a montagem de um robô modular (Passo 1, acima), sugere-se a sequência de etapas apresentadas na Figura 5.1.

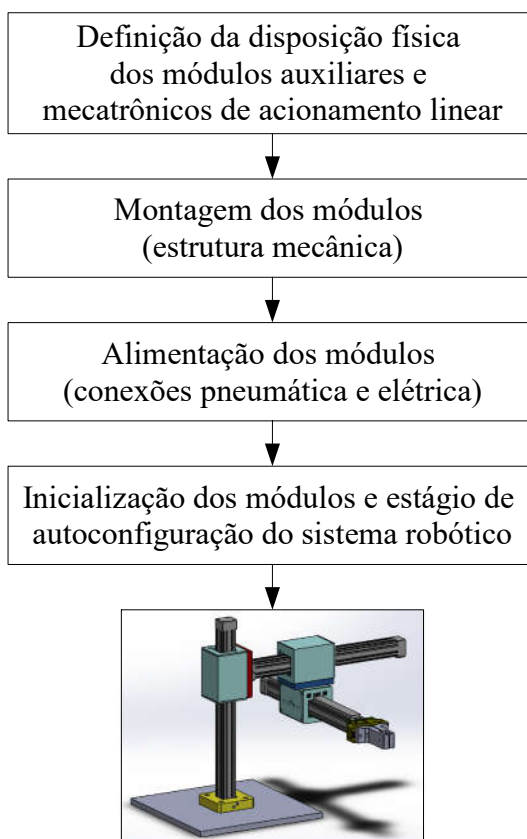


Figura 5.1 – Sequência de montagem de um robô modular

Como ilustrado na Figura 5.1, uma vez que os elementos mecânicos são selecionados, é necessário realizar a montagem física destes dispositivos. As fixações destas unidades, como apresentado na Seção 4.1, são realizadas de modo que o 1º e o 2º GDL são, por meio de um

módulo auxiliar, conectados entre uma base fixa e a conexão de um módulo mecatrônico, de acordo com as figuras 4.3 e 4.5.

A alimentação do robô modular baseia-se no fornecimento de ar comprimido e energia elétrica para cada módulo mecatrônico. Como cada unidade possui uma interface de rede (Figura 4.4), é necessário conectar cada uma das unidades a uma rede local, visando a estabelecer comunicação entre as unidades mecatrônicas e, conseqüentemente, do sistema Multiagente.

Na inicialização do sistema, cada unidade (módulo mecatrônico) disponibiliza informações sobre suas habilidades, características físicas, tipo de acionamento e localização na própria estrutura modular, entre outras. Estas características são obtidas através dos agentes *recursos* e *cognitivos* que operam no módulo mecatrônico.

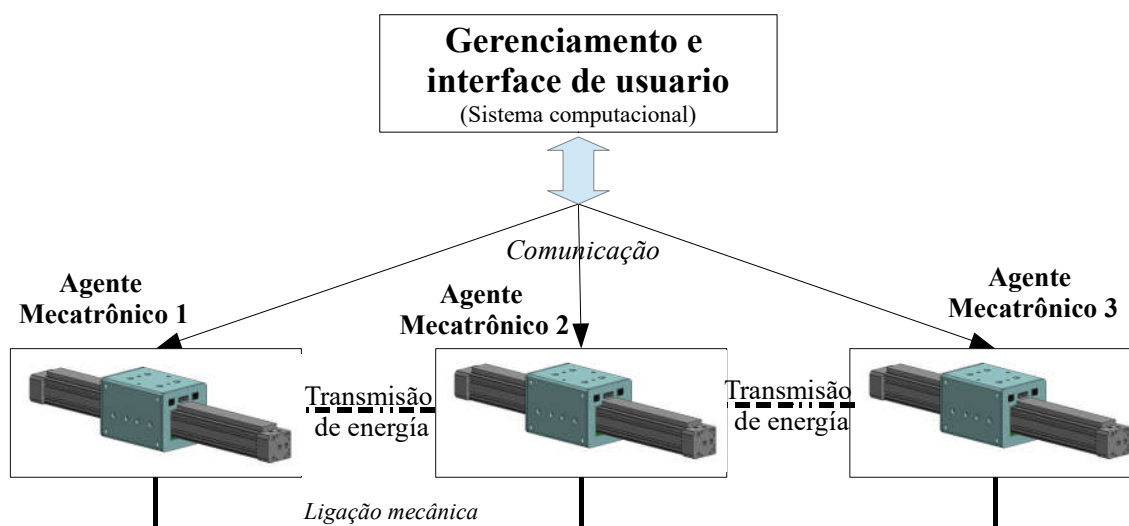


Figura 5.2 – Componentes de uma plataforma de um robô modular

Neste trabalho, propõe-se que os módulos mecatrônicos que compõem o robô sejam conectados a uma mesma rede local composta por um sistema robótico formado pelos módulos mecatrônicos e por um computador que serve como interface com o usuário e onde são executados os agentes que realizam o gerenciamento do sistema. Dessa forma, um sistema computacional é responsável tanto pela interface com o usuário como pela execução dos agentes do sistema Multiagente, permitindo a inserção dos pontos chaves da trajetória, a realização de ajustes dos parâmetros de controle e a visualização dos estados dos módulos ativos e dos agentes que estão interagindo no sistema. Na Figura 5.2 apresenta-se os componentes presentes na rede local e o tipo de ligação que existe entre os componentes de uma plataforma de um robô modular (comunicação, transmissão de energia e ligação mecânica).

5.2 Estudo de caso de um robô cartesiano de acionamento pneumático

Após a montagem e a inicialização do sistema, o sistema Multiagente irá executar dois algoritmos necessários para o funcionamento do robô: o algoritmo de planejamento de trajetória e o algoritmo de controle de posição do manipulador. Com a finalidade de apresentar o princípio de funcionamento destes algoritmos, são apresentados a seguir os resultados de uma trajetória arbitrária (deslocamento ponto a ponto) considerando a cinemática de um robô cartesiano e a simulação do controle para o seguimento da trajetória gerada.

5.2.1 Simulação de Planejamento de trajetória

Para a execução do algoritmo de planejamento de trajetória, é necessário que os pontos intermediários (pontos chaves ou *key-points*) sejam definidos precisamente em função da tarefa específica a ser efetuada. Estes pontos são importantes para que o robô se mova no caminho tridimensional a ser percorrido. Considerando um robô cartesiano de 3 GDL, as coordenadas de cada ponto em relação aos eixos x , y e z resultam, para cada nó, em um vetor do tipo:

$$p_i = (x_i, y_i, z_i) \quad i = 1, \dots, N \quad (5.1)$$

onde N é o número de pontos intermediários da trajetória

Para o estudo de caso, foi implantada uma rotina computacional em Matlab® que permite, para cada junta do robô, a geração da trajetória (*spline* 3ª ordem) em função dos parâmetros desejados, resultando nos intervalos de tempo entre os nós subsequentes (segmento h_i) e os valores de posição dos nós de passagem no espaço tridimensional (pontos intermediários). Estes parâmetros são necessários para que todas juntas iniciem e terminem o movimento de forma sincronizada, respeitando os tempos desejados para o movimento do robô. O processo de planejamento de trajetória pode ser sintetizado por meio das etapas apresentadas na Figura 5.3.

A partir da informação dos pontos intermediários e o tempo entre dois pontos intermediários (parâmetro h_i) da trajetória, é realizada a validação dos pontos. Esta etapa objetiva avaliar se os pontos informados encontram-se dentro do espaço de trabalho do robô, que, considerando as dimensões limites do efetuator, são determinados previamente por meio de uma função do agente líder de coalisão.

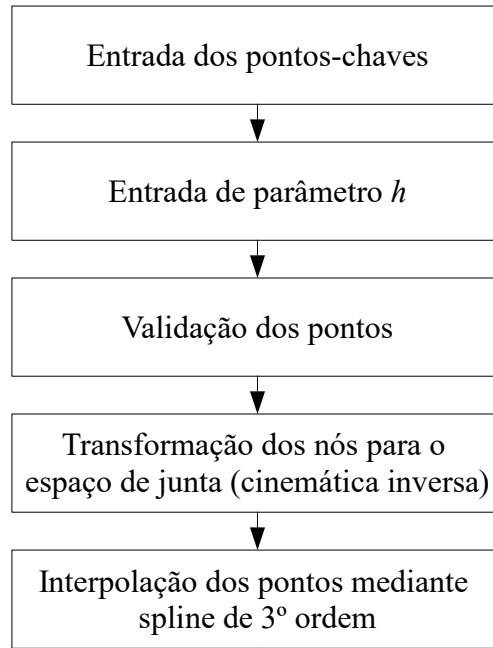


Figura 5.3 – Processo de geração de trajetória para robôs modulares

Os algoritmos que determinam a cinemática inversa, apresentados na Seção 3.2, são utilizados na transformação dos valores dos nós, visto que transformam as coordenadas dos pontos do espaço cartesiano para o espaço das juntas. Esta etapa é de extrema importância nesta abordagem, já que o processamento do controle e a interpolação dos pontos são realizadas no espaço de juntas, possibilitando que o agente líder de coalisão, entidade que processa estes algoritmos, envie as referências de movimentação para cada uma das juntas (módulos mecatrônicos). A execução destes algoritmos tem como resultado um conjunto de pontos para cada uma das juntas prismáticas e um tempo de percurso (h), os quais são interpolados através das splines de terceira ordem, descritas na Seção 4.4.

O processo de interpolação resulta, para cada um dos atuadores, em um vetor que relaciona os valores de posição do atuador com a variável de tempo. O intervalo de tempo, no qual o efetuador do robô percorrerá o caminho completo para a trajetória desejada, é expresso em função do valor do parâmetro h_i e do número de pontos intermediários definidos pelo programador para o cálculo da trajetória. Assim, o tempo total da trajetória é dado por:

$$(N-1) h_i \quad i = 1, \dots, N-1, \quad (5.2)$$

sendo N o número de pontos intermediários e h_i o tempo entre os pontos nos instantes de tempo t_i e t_{i-1} .

Para exemplificar o processo de geração de trajetórias utilizando a metodologia desenvolvida, é apresentado um caso no qual o efetuator do robô deve percorrer um caminho formado por cinco pontos localizados em seu volume de trabalho. As coordenadas dos pontos intermediários são apresentadas na Tabela 5.1, enquanto que a Figura 5.4 apresenta estes pontos e a trajetória no espaço de trabalho do robô.

Tabela 5.1 – Coordenadas dos pontos do efetuator

| Ponto | $x(m)$ | $y(m)$ | $z(m)$ |
|-------|--------|--------|--------|
| P0 | 0,12 | 0,1 | 0 |
| P1 | 0,0942 | 0,098 | 0,0567 |
| P2 | 0,1542 | 0,1467 | 0,193 |
| P3 | 0,3 | 0,1028 | 0,2999 |
| P4 | 0,32 | 0,32 | 0,32 |

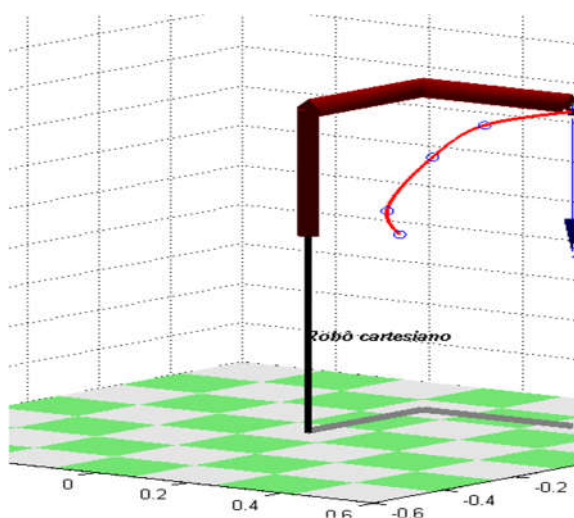


Figura 5.4 – Pontos intermediários e respectiva trajetória de 3º grau.

Neste estudo, considera-se que os valores de cada segmento (h_i) são $t_i=[0; 2,5795; 5,3923; 6,8803; 8,0044]$. As figuras 5.5, 5.6 e 5.7, apresentam a posição de cada junta do atuador em função do tempo, bem como os tempos correspondentes ao percurso total da trajetória.

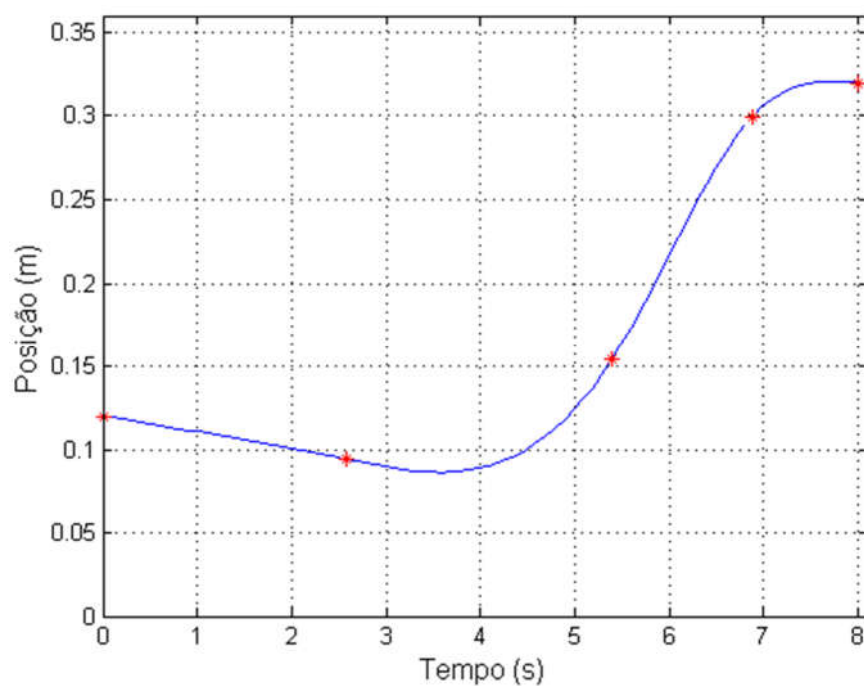


Figura 5.5 – Curvas de posição da junta 1 do robô modular

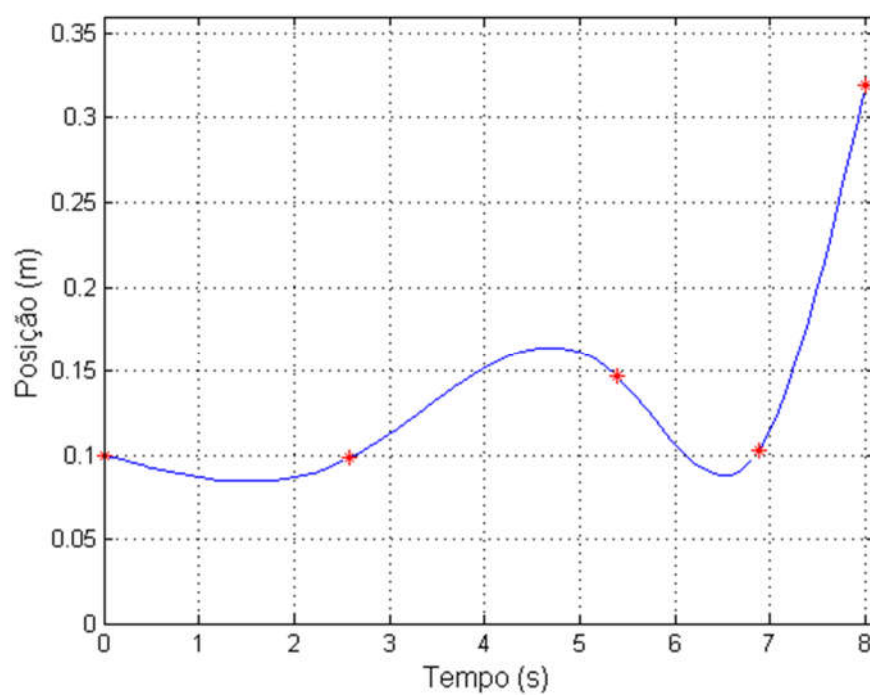


Figura 5.6 – Curvas de posição da junta 2 do robô modular

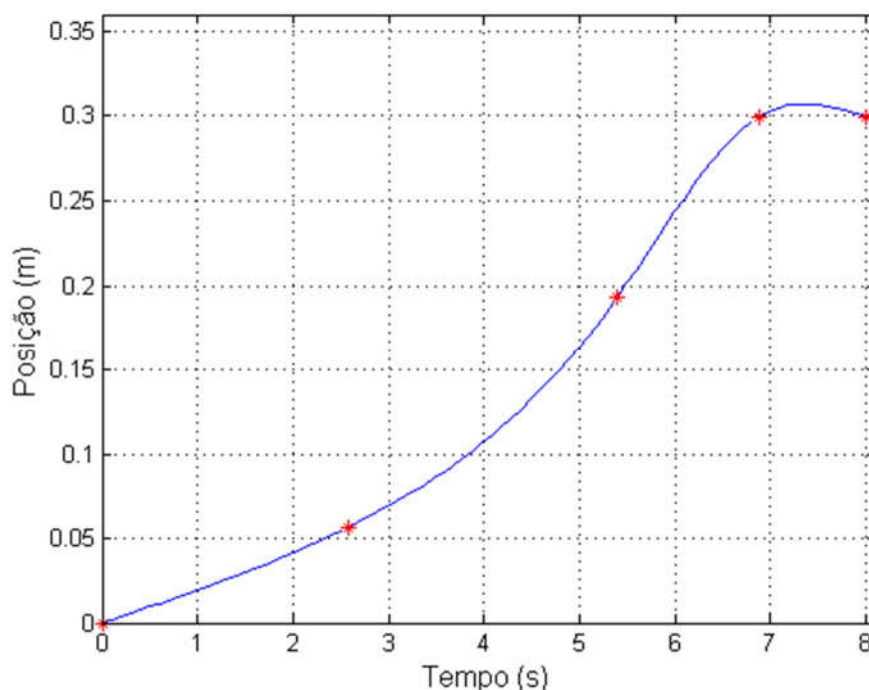


Figura 5.7 – Curvas de posição da junta 3 do robô modular

Os pontos intermediários, destacados em vermelho, são introduzidos manualmente pelo usuário. A trajetória gerada, à qual se refere a interpolação dos pontos intermediários, é determinada mediante funções polinomiais de terceiro grau, garantindo, assim, a continuidade da função do movimento até as grandezas de velocidade e aceleração.

5.2.2 Controle aplicado ao seguimento de uma trajetória

Esta seção apresenta os resultados obtidos na simulação de um robô cartesiano de 3 GDL, executada por meio do modelo matemático apresentado na Seção 4.3. A trajetória de referência baseia-se nos estudos apresentados na Seção 5.2.1. Seguindo esta estratégia, a referência de entrada para o algoritmo de controle é gerada a partir da predefinição dos pontos da trajetória. Conforme já discutido, o algoritmo de controle utilizado baseia-se no método de controle por Torque Calculado, acrescido de termos de compensação de atrito nos atuadores do robô [Sarmanho, 2014]. Os ganhos utilizados no controlador proposto foram sintonizados para obter um seguimento de trajetória com menor erro, procurando minimizar os efeitos oscilatórios e de alta frequência que apresentam os sinais de controle. A Tabela 5.2 apresenta os ganhos utilizados nos procedimentos de simulação do controlador. É importante destacar que as sintonias dos ganhos, tanto para simulações como para aplicações experimentais, é ainda um campo de pesquisa em aberto, pois os ganhos dependem de fatores, tais como, geometria, atrito,

carga deslocada, dentre outros. Existem trabalhos que visam a superar estas dificuldades, utilizando, por exemplo, técnicas adaptativas, auto ajuste de ganhos, redes neurais [Aziz e Bone, 1998; Cheng et al., 2014, Gervini, 2014], dentre outras, que poderão ser abordados em trabalhos futuros.

Tabela 5.2 – Ganhos do controlador TC [adaptado de Sarmanho, 2014]

| Ganho | 1GDL | 2GDL | 3GDL |
|--------------|-------------|-------------|-------------|
| K_{di} | 80 | 80 | 80 |
| λ_i | 20 | 20 | 30 |
| K_{ai} | 20 | 40 | 60 |

Em relação aos resultados do algoritmo de controle, são apresentados os gráficos com resultados relacionados ao posicionamento das juntas durante a movimentação do robô, levando-se em conta as trajetórias determinadas na Seção 5.2.1. É importante destacar que, no presente estágio, o trabalho não objetiva avaliar o desempenho do sistema de controle.

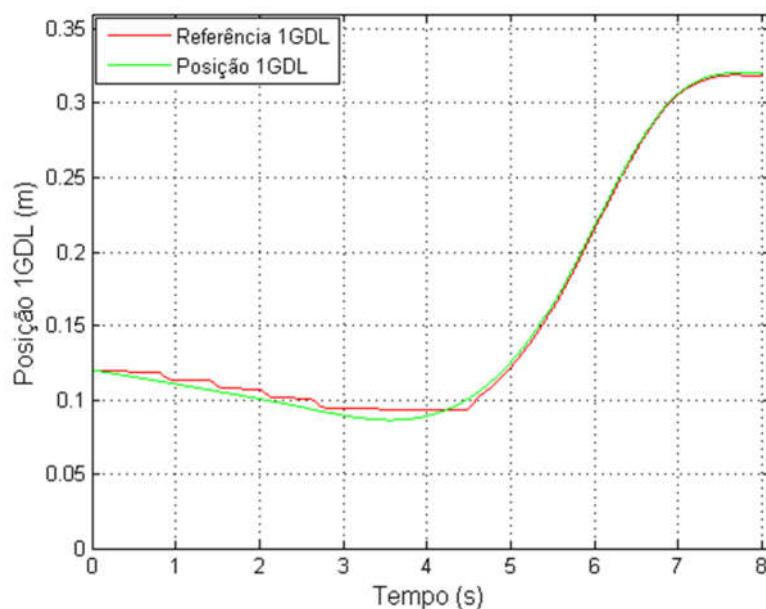


Figura 5.8 – Controle Torque Calculado 1º GDL

A Figura 5.8 apresenta os resultados relativos ao 1º GDL do modelo de simulação do robô cartesiano de 3 GDL, considerando o controlador por Torque Calculado no seguimento de trajetória do tipo ponto a ponto.

As figuras 5.9 e 5.10 apresentam os resultados das simulações relativas ao 2º e 3º GDL, respectivamente.

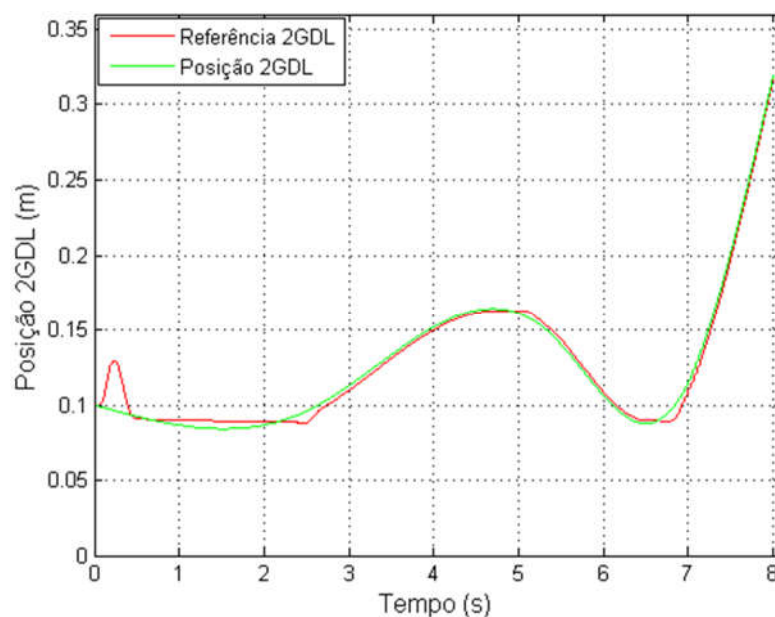


Figura 5.9 – Controle Torque Calculado 2º GDL

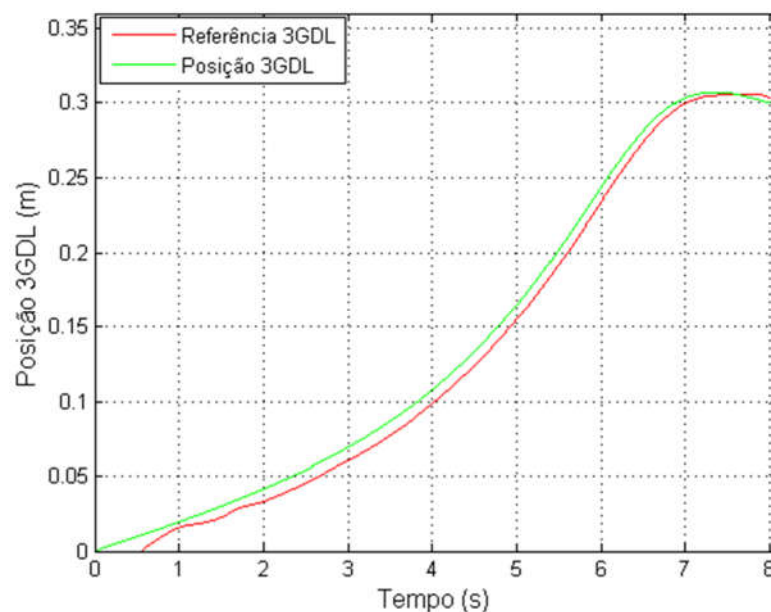


Figura 5.10 – Controle Torque Calculado 3º GDL

Como já mencionado, os robôs modulares são gerenciados por um sistema Multiagente, que é responsável pelo reconhecimento do tipo de estrutura cinemática, pela execução do algoritmo de controle, planejamento de trajetória e de interação com os módulos mecatrônicos do sistema. Isto é realizado através de um protocolo de comunicação específico para cada tarefa

do robô. Nesse sentido, para atender as funcionalidades de um robô modular, o algoritmo de controle foi desenvolvido em função das informações associadas a cada módulo mecatrônico, as quais devem ser predefinidas antes da inicialização do sistema. No caso do robô modular cartesiano de 3 GDL, os três módulos mecatrônicos apresentam características físicas similares. Assim, o comprimento do elo, a massa de cada módulo e as orientações relativas das juntas, conforme apresentado nas seções 4.2 e 4.3, são consideradas similares.

5.2.3 Gerenciamento de robôs segundo o conceito de Sistemas Multiagentes

Além das tarefas específicas relacionadas ao planejamento de trajetória e de controle do robô, é importante descrever como se dá a organização do sistema modular, isto é, as sequências de atividades que as entidades computacionais (agentes) realizam desde a montagem física do robô modular, até o momento da realização de uma tarefa solicitada pelo usuário. A Figura 5.11 ilustra as etapas de inicialização do sistema Multiagente e o gerenciamento do robô após a montagem dos módulos mecatrônicos.

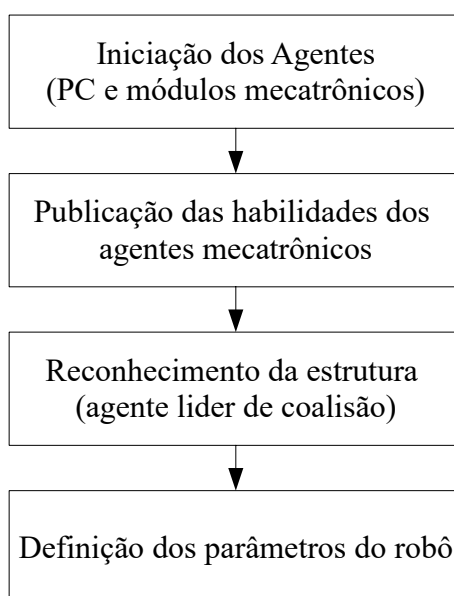


Figura 5.11 – Inicialização do sistema computacional de gerenciamento

Após a montagem e alimentação do robô modular, são inicializadas as entidades do sistema, isto é, os agentes que operam em diferentes unidades da mesma rede (agentes presentes no computador e nos módulos mecatrônicos). Inicialmente, com base no protocolo FIPA, apresentado na Seção 3.4, as entidades publicam suas habilidades no agente de páginas amarelas (DF) para que estejam disponíveis para outros agentes, permitindo, assim, que o

agente líder de coalizão utilize estas informações, reconhecendo todas as unidades mecatrônicas e o tipo de estrutura montada. Dentre as informações disponibilizadas, as entidades informam a sua localização dentro da estrutura juntamente com suas características físicas, possibilitando que o agente líder de coalizão reconheça o tipo de estrutura formada e defina os parâmetros do robô relativos à cinemática e à dinâmica, como apresentado na Seção 4.3. Por fim, é importante mencionar que esta reconfiguração deve ser realizada toda vez que o sistema é inicializado, ou quando é montado. As unidades de montagens anteriores podem ser reutilizadas, seguindo este mesmo procedimento.

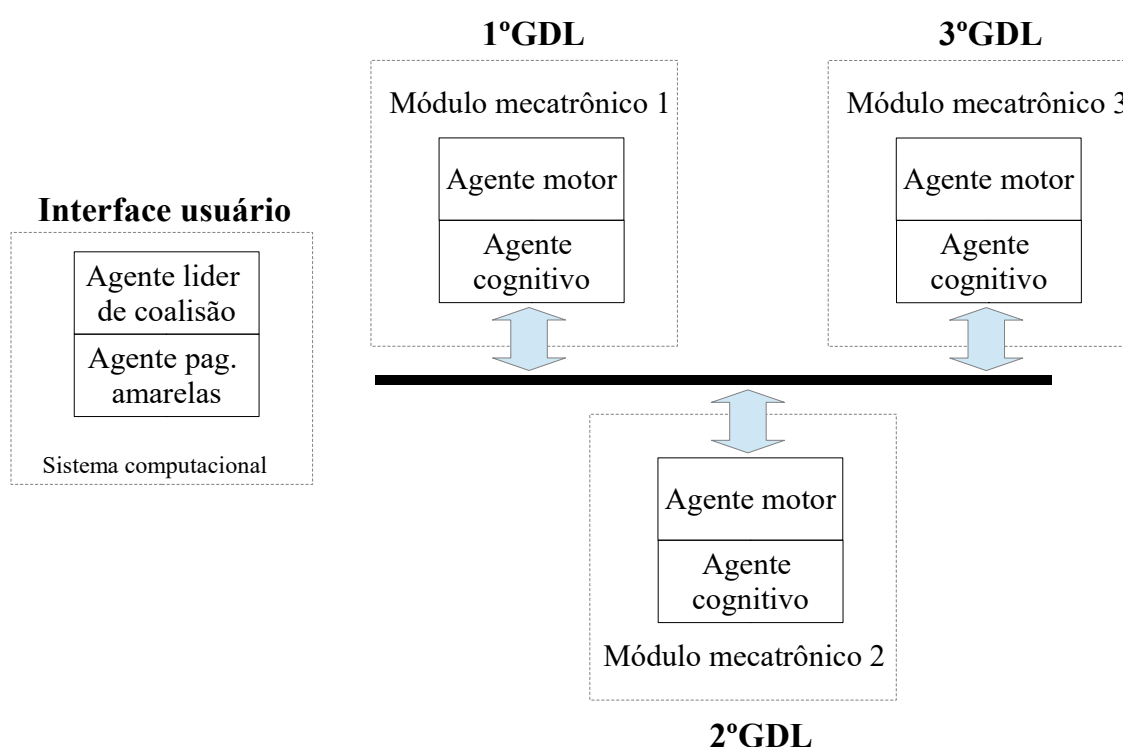


Figura 5.12 – Localização dos agentes no sistema

É importante mencionar que os agentes são entidades que não migram dentro da rede, e que cada entidade possui lugar e tarefa definida. No sistema computacional são iniciadas as rotinas de interface de usuário e são executados o agente líder de coalizão e o agente páginas amarelas. Os agentes recursos e os agentes de distribuição são executados nos módulos mecatrônicos. Os agentes executados nos módulos mecatrônicos, que são do tipo cognitivo e agente motor, são encarregados dos processamentos dos algoritmos de controle, leitura e atuação dos sensores, respectivamente. Uma representação destas entidades é apresentada na Figura 5.12.

Uma interface gráfica para simulação foi programada visando permitir visualizar como será a inicialização do sistema com a utilização de Netbeans, similar a plataforma IADE utilizada por Cavalcante, 2012. Esta rotina, através dos procedimentos associados à montagem do robô, recebe as informações e apresenta os módulos que estão ativos e conectados à estrutura. Além disso, é possível introduzir os pontos-chaves para o planejamento de trajetória e ajuste dos ganhos de controle de cada módulo, tal como apresentado nas figuras 5.13 e 5.14.



Figura 5.13 – Indicação dos módulos mecatrônicos que estão conectados à plataforma

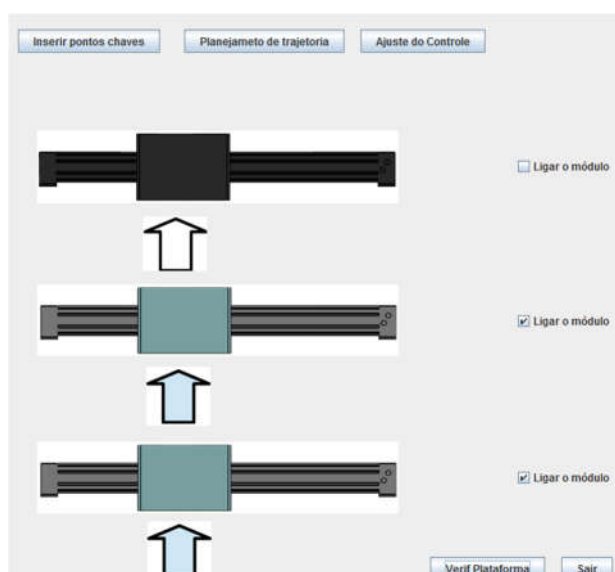


Figura 5.14 – Módulos ativos conectados no sistema

O sistema identifica cada módulo mecatrônico que está devidamente conectado e iniciado e o apresenta com uma cor diferente. Estas rotinas foram programadas ambiente virtual Java Netbeans, assim como a interface onde são inseridos os pontos chaves da trajetória e tipo de trajetória a ser executada.

As operações de planejamento de trajetória e controle são funções instanciadas pelo agente líder de coalisção que encontra-se no computador principal. As informações referentes às

trajetórias e à atuação são repassadas pelo “comportamento principal” a cada unidade mecatrônica para a sua execução. As abrangências dos principais tipos de agentes estão esquematizadas na Figura 5.15.

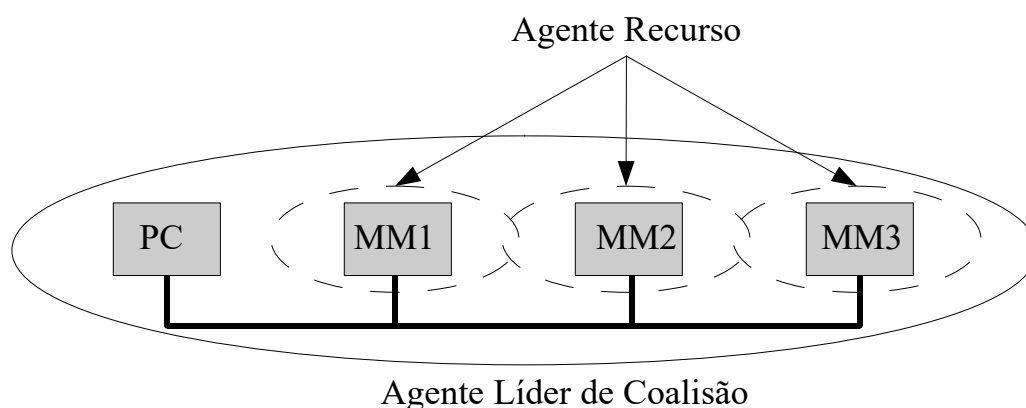


Figura 5.15 – Abrangência e localização dos agentes da plataforma

A Figura 5.15, apresenta o sistema Multiagentes, onde cada bloco é uma entidade física (um computador ou um módulo mecatrônico) e as elipses contínuas representa o agente líder de coalisção, cuja abrangência é em todo o sistema robótico. Já os agentes recurso (ou agente mecatrônico), composto por agentes cognitivos e agentes motores, tem uma abrangência menor, e somente de um módulo mecatrônico, que aqui se representa por elipses descontínuas.

É ainda importante destacar no âmbito do estudo de caso, que todas as simulações (planejamento de trajetória, controle e gerenciamento das unidades através do uso de uma plataforma Multiagente) foram executadas separadamente. Em trabalhos futuros, todas estas ações deverão ser conjugadas em um sistema integrado, que permita a montagem e operação de um robô modular, de acordo com os objetivos gerais da presente pesquisa.

5.3 Considerações finais

Basicamente, os robôs modulares são constituídos por módulos mecatrônicos conectados por módulos auxiliares. Os módulos mecatrônicos são os elementos principais, possuindo capacidade de acionamento das juntas e de processamento computacional. Um dos aspectos importantes na concepção módulos mecatrônicos consiste da capacidade de tomada de decisões na execução das tarefas requeridas. Nesta perspectiva, no âmbito do presente trabalho, os módulos mecatrônicos foram concebidos com características de agentes computacionais, de modo que estas entidades são responsáveis pela coordenação e pela

organização dos recursos dos módulos mecatrônicos, características que permitem a obtenção de robôs modulares de maneira rápida e eficaz.

Levando-se em conta as características apresentadas em relação à construção de robôs modulares, é possível definir no presente contexto um *módulo mecatrônico* como uma *entidade eletromecânica com a capacidade de realizar algum tipo de movimentação, bem como coordenar e definir movimentos mais complexos em suas tarefas*.

A interação entre os módulos mecatrônicos baseia-se na comunicação que os agentes estabelecem por meio de um protocolo bem definido (FIPA). Assim, o gerenciamento das unidades mecatrônicas consiste na formação de um sistema Multiagente, que tem como objetivo principal a formação “lógica” e física de um robô manipulador incluindo todas as funções necessárias para sua adequada operação incluindo o controle de seu movimento.

Para garantir o funcionamento dos robôs modulares, os módulos mecatrônicos devem apresentar características próprias dos agentes, que, no escopo do presente trabalho, foram definidas como:

Autonomia: Os módulos mecatrônicos têm a capacidade de atuar sem intervenção externa ou de outros agentes. Desta forma, mesmo operando isoladamente, isto é, sem interação com outros agentes, cada módulo pode executar suas ações de forma autônoma (Figura 5.16), consequentemente cada módulo mecatrônico tem a capacidade de reagir a um estímulo a través das execuções de uma ou mais de suas funções com comportamentos predefinidos.

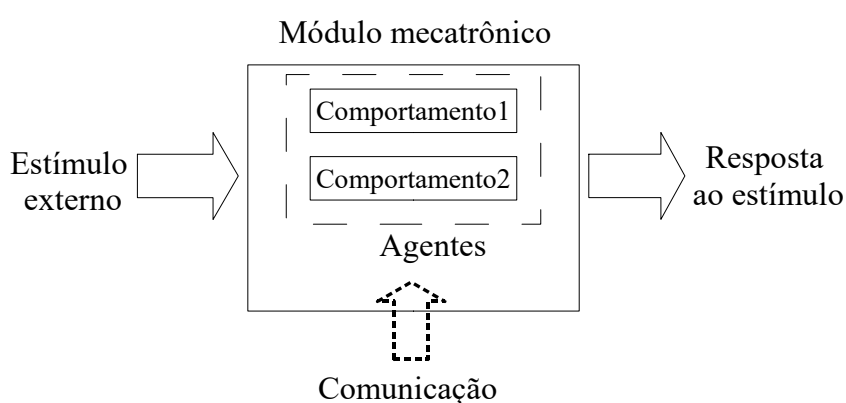


Figura 5.16 – Comportamentos dos módulos mecatrônicos

Um exemplo de autonomia de módulos mecatrônicos consiste na execução de ações relativas a um comportamento cíclico que o agente cognitivo executa chamando a função que executa o algoritmo de controle. Desta forma, mesmo que o sistema não receba uma referência externa, ele opera com uma referência interna, cuja finalidade é a de manter o sistema parado (ou sem movimento) na posição considerada como “zero”, tal como apresentado na Figura 5.17.

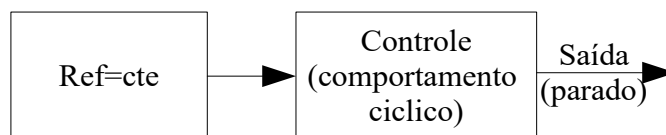


Figura 5.17 – Comportamento cíclico de um agente (execução de um algoritmo de controle)

Sociabilidade: é a capacidade que tem o módulo mecatrônico de interagir com outros similares por meio de uma linguagem de comunicação. O Jade utiliza o protocolo de comunicação FIPA, que os sistemas computacionais e os módulos mecatrônicos utilizam para realizar as negociações associadas à movimentação das juntas do robô. Uma representação desta interação é apresentada na Figura 5.18.

A negociação apresentada na Figura 5.18 é um caso típico para a atualização das referências da movimentação de uma junta. O agente líder de coalisção solicita um movimento ponto a ponto a um módulo mecatrônico; como consequência, a trajetória desejada que o sistema de controle utiliza para o deslocamento é atualizada, como apresentado na Figura 5.17.

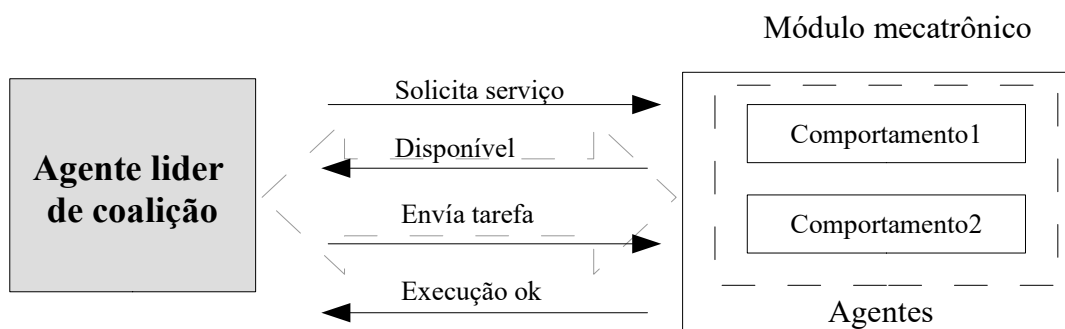


Figura 5.18 – Negociação entre um módulo mecatrônico e um computador

Reatividade: ocorre quando um módulo mecatrônico reage em um tempo estabelecido a estímulos que recebe do seu meio. Assim, considerando o controlador aplicado a uma operação de posicionamento, o módulo mecatrônico reage diante dos esforços externos que poderão levar o efetuador a uma posição diferente da posição desejada. Este princípio é similar ao comportamento de autonomia, apresentado na Figura 5.16. Neste caso, independentemente do fato de o módulo mecatrônico se encontrar ou não em movimento, o comportamento cíclico que apresenta um agente permite realizar as correções necessárias por intermédio do algoritmo de controle, de forma que o erro de posição seja corrigido, levando o efetuador à posição desejada, tal como apresentada na Figura 5.19.

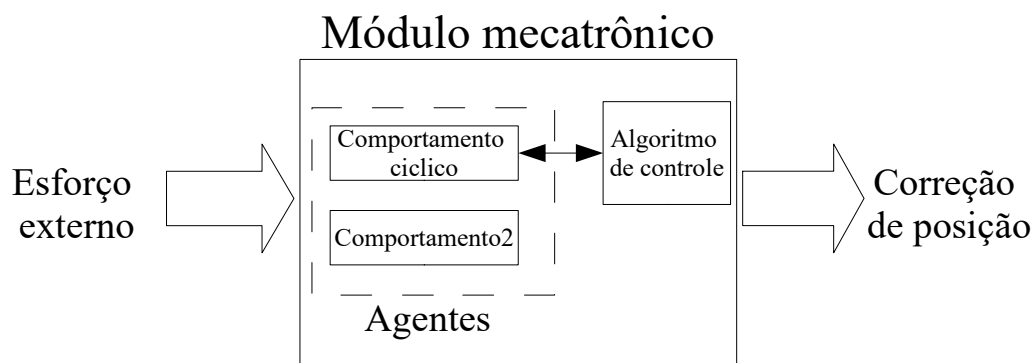


Figura 5.19 – Comportamento reativo de um módulo mecatrônico

Iniciativa: Um módulo mecatrônico além de reagir ante mudanças que, porventura, seu meio apresenta, deve ter comportamentos com caráter empreendedor, tomando a iniciativa de reagir visando a satisfazer os seus objetivos inicialmente estabelecidos. Por exemplo, quando o módulo mecatrônico é inicializado, nem sempre o efetuator está na posição inicial definida como “zero” do sistema mecatrônico. Neste caso, embora não exista uma excitação externa, o controle com comportamento cíclico do agente comanda o módulo mecatrônico de forma a que ele corrija a sua posição até que se encontre no ponto convencionado para a inicialização. Este comportamento está representando na Figura 5.19, em uma situação na qual não existem esforços externos.

Finalmente, baseadas nas características que apresenta cada unidade elementar (módulo mecatrônico), o robô formado terá características similares como apresentadas anteriormente. Por outra parte, para a utilização de esta metodologia e realizar a configuração de uma estrutura modular robotizada, e finalmente a execução de uma determinada tarefa, é necessário ter em consideração a sequencia apresentada na Figura 5.20.

De acordo com a Figura 5.20, antes de realizar qualquer montagem, é necessário estabelecer o tipo de geometria robótica necessária para a realização da tarefa requerida. Definida esta etapa, procede-se a montagem dos módulos mecatrônicos e os módulos auxiliares para a formação do robô, e seguidamente, a sua alimentação. Concluída a etapa de montagem manual, é necessária a inicialização do sistema multiagente, para o reconhecimento da geometria do robô e definição da cinemática direta e inversa. Finalmente, o usuário deve definir o tipo de tarefa a ser realizada e quais são os pontos intermediários de passagem do efetuator.

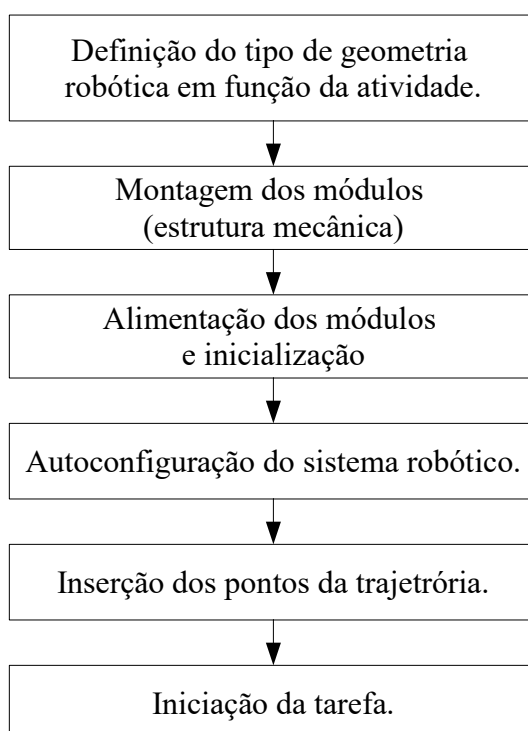


Figura 5.20 – Fluxograma de utilização da metodologia para a configuração de robôs modulares.

6 CONCLUSÕES FINAIS E TRABALHOS FUTUROS

6.1 Conclusões

O presente trabalho pretende contribuir no âmbito do desenvolvimento tecnológico industrial, apresentando uma proposta baseada nas técnicas de inteligência artificial distribuída, neste caso, teoria de agentes, para permitir a montagem sequencial de manipuladores industriais. Nesse sentido, a partir das propriedades dos agentes mecatrônicos, são definidos os requisitos necessários para a utilização de sistemas modulares na montagem de robôs manipuladores de uso industrial, atendendo às características de auto-organização da estrutura robótica. Para validar o método proposto, é apresentado um procedimento para o desenvolvimento de um robô modular cartesiano de 3 GDL, analisando aspectos de controle de posicionamento, de geração de trajetória e do gerenciamento do sistema Multiagente, os quais mostram que os robôs modulares podem ser viáveis quanto à sua implementação.

Na manufatura, cada vez mais, a dinâmica dos sistemas produtivos exige melhorias em diversos níveis do processo, como, por exemplo, as linhas de montagem, as máquinas, os robôs, entre outras. Para atender estas necessidades, diversos estudos vêm sendo desenvolvidos. Atualmente, com o surgimento de novas aplicações dos conceitos de modularidade e das pesquisas baseadas em agentes, o conceito de sistemas evolutivos tem permitido o desenvolvimento de sistemas físicos que se caracterizam pela autonomia, sociabilidade, reatividade, iniciativa, entre outros, na execução das ações e/ou tarefas, apresentando um cenário apropriado para a formação de robôs modulares. Ademais, observa-se que na literatura existem poucas abordagens que tratam dos conceitos relacionados ao projeto e desenvolvimento de manipuladores robóticos a partir da utilização de módulos mecatrônicos. Nas pesquisas realizadas, constata-se que alguns autores pesquisam somente sobre a montagem de robôs modulares, enquanto que outros estudam a utilização de sistemas Multiagentes em módulos mecatrônicos em outras aplicações.

Existem diversas questões que devem ser analisadas quanto à utilização de sistemas modulares na montagem de robôs modulares. Os aspectos discutidos neste trabalho, como os requisitos de construção e os algoritmos computacionais (características mecânicas dos módulos mecatrônicos, montagem do manipulador, identificação da estrutura mecânica, funções de cálculo da cinemática e dinâmica da estrutura, agentes, gerenciamento dos módulos mecatrônicos, entre outros), mostram que, apesar das funcionalidades mencionadas serem

analisadas e validadas separadamente, os robôs modulares têm potencial para implementação em aplicações industriais, visto que a tecnologia e os conhecimentos necessários para desenvolver protótipos experimentais estão disponíveis. Nesta perspectiva, entende-se que, para validar todas as funcionalidades robôs modulares em ambientes virtuais, é necessário simular estes sistemas em plataformas que reproduzam todas estas características de forma conjunta, isto é, levando-se em conta os aspectos de movimentos relativos das juntas, de construção dos módulos, de cinemática e de dinâmica da estrutura, assim como os algoritmos computacionais dos agentes.

Considerando os objetivos parciais deste trabalho, pode-se inferir que: 1) os módulos mecatrônicos são alternativas viáveis na formação de manipuladores, sempre o gerenciamento do robô seja um sistema colaborativo, como os sistemas Multiagentes; 2) os requisitos de projetos (mecânicos, eletrônicos e eletropneumáticos) de um módulo mecatrônico são definidos para o caso particular de um sistema pneumático de acionamento linear, mesmo que as especificações nos casos de sistemas com acionamento elétrico e hidráulico sejam similares; 3) algoritmo de identificação da arquitetura do manipulador e os cálculos da cinemática direta e inversa apresentaram resultados satisfatórios para ambientes de simulação virtual; 4) os algoritmos de controle propostos indicam comportamentos satisfatórios em relação às características funcionais dos módulos mecatrônicos de acionamento pneumático. No entanto, considerando o modelo do robô modular, ou seja, a estrutura formada por módulos mecatrônicos, são necessários ajustes em implementações futuras, como, por exemplo, a utilização de estratégias de controle considerando o atrito e a compensação da gravidade; e, por fim, 5) as funções de geração de trajetória, implementadas por splines de 3ª grau, demonstram-se adequadas para as funcionalidades dos robôs modulares, já que possibilitam movimentos suaves, e são de fácil implementação, possibilitando aplicações em um protótipo funcional;

Os robôs modulares, por serem reconfiguráveis e apresentarem características apropriadas para trabalhar em conjunto com outros tipos de equipamentos, são alternativas viáveis para aplicações em indústrias com linhas de manufatura flexíveis e evolutivas. No entanto, para implementações práticas destes sistemas, é necessário realizar uma abordagem mais aprofundada dos conceitos discutidos, permitindo, assim, em pesquisas futuras, a implantação e utilização generalizada deste tipo de sistema.

6.2 Trabalhos futuros

Seguindo a linha de pesquisa deste trabalho e tendo em vista os resultados obtidos, trabalhos futuros deverão ser realizados, dentre os quais pode-se destacar:

- Montagem experimental de um sistema modular com capacidade de processamento de agentes mecatrônicos, similares aos propostos neste estudo;
- Definição dos agentes motores compatíveis com o sistema proposto, possibilitando a utilização da plataforma proposta com um hardware com diferentes tipos de configuração e acionamento;
- Programação de novos agentes cognitivos para o gerenciamento dos algoritmos de controle com capacidade de processamento em tempo real e controle distribuído;
- Testar outros algoritmos de controle;
- Realizar a avaliação dos algoritmos que reconhecem a cinemática e dinâmica utilizando sistemas com outras configurações geométricas.

REFERÊNCIAS BIBLIOGRÁFICAS

AFNOR, Groupe - Normalisation, Certification, Edition et Formation. França, 2016. Acessado em Julho de 2016. www.afnor.org/

Atta-Konadu, R.; Lang, S.Y.T.; Zhang, C.; Orban, P.. **Design of a robot control architecture**. pp. 1363–1368 Vol. 3, 2005.

Aziz, S.; Bone, G.M. **Automatic tuning of an accurate position controller for pneumatic actuators**. In IEEE/RSJ International Conference on Intelligent Robots and Systems. Proceedings, pp. 1782–1788 vol.3, 1998.

BARA, **British Automation & Robot Association**. Acessado em Julho de 2016. www.bara.org.uk/

Barata, José Antônio. **Coalition based approach for shop floor agility – a multiagent approach**. Universidade Nova de Lisboa, Portugal. Tese de doutorado. 2003.

Barrientos, Antonio. **Fundamentos de robótica**. ISBN 978-84-481-0815-1. Editora McGraw-Hill. Espanha, 1997.

Biagiotti, Luigi; Melchiorri, Claudio. **Trajectory Planning for Automatic Machines and Robots**. Editora Springer Berlin Heidelberg, 2009.

Bobrow, J.E.; McDonell, B.W. **Modeling, identification, and control of a pneumatically actuated, force controllable robot**. IEEE Transactions on Robotics and Automation 14, 732–742. 1998.

Buzzetto, Fabiano Alberto. **Implantação de uma nova sistemática de gerenciamento de projetos em uma empresa de componentes eletrônicos**. Dissertação de mestrado em Programa de Pós-Graduação em Engenharia de Produção UFRGS, Porto Alegre, 2008.

Cavalcante, A.L.D. (2012). **Arquitetura baseada em agentes e auto-organizável para a manufatura**. Tese de doutorado em Programa de Pós-Graduação em Engenharia Elétrica. UFRGS. Porto Alegre, 2012.

Chapra, Steven C.; Canale, Raymond P. **Métodos Numéricos para Engenharia**. Editora McGraw-Hill. ISBN 978-85-86804-87-8. 2008.

Chen, Ming; Yang, Guilin. **Configuration independent kinematics for modular robots**. In IEEE International Conference on Robotics and Automation, Proceedings, pp. 1440–1445 vol.2. 1996.

Cheney, Ward; Kincaid, David. **Numerical Mathematics and Computing**. Editora Cengage Learning. ISBN 978-1-133-10371-4. 2012.

Cheng, X.; Zhang, S.; Chen, Y.; Zhang, H. **Auto-commissioning and adaptive tuning of servo control parameters in an electro-hydraulic system based on physical plant model**.

In IEEE 15th Workshop on Control and Modeling for Power Electronics (COMPEL), pp. 1–5. 2014.

Craig, Jhon J. **Introduction to Robotics: Mechanics and Control**. Editora Pearson/Prentice Hall. 2005.

Cukla, Anselmo Rafael. **Arquitetura microcontrolada programável aplicada ao controle de um servoposicionador pneumático**. Dissertação de mestrado. Programa de pós-graduação em Engenharia Mecânica UFRGS. Porto Alegre. 2012.

Ferreira, J.; Ribeiro, L.; Neves, P.; Akillioglu, H.; Onori, M.; Barata, J. **Visualization tool to support multi-agent mechatronic based systems**. In IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, pp. 4372–4377. 2012.

FESTO, Festo Corporation. Acessado em Julho de 2016. www.festo.com.

Fialho, Arivelto Bustamante. **Automação Pneumática: Projetos, Dimensionamento e Análise de Circuitos (2ª Edição)**. Editora Erica, ISBN 9788571949614. 2004.

FIPA, **Agent Communication Language Specifications**. 2002. Acessado em Julho de 2016. <http://www.fipa.org/specs/fipa00061/index.html>

FIPA, Specification. 2002. Acessado em Julho de 2016. <http://www.fipa.org/specs/fipa00061/index.html>.

Frasson, Marcelo. **Projeto de um robô cartesiano com acionamento pneumático**. Dissertação de mestrado. Programa de pós-graduação em Engenharia Mecânica UFRGS, Porto Alegre, 2007.

Fu, K.S.; González, R.C.; Lee, C.S.G. **Robotics: control, sensing, vision, and intelligence**. Editora McGraw-Hill. 1987.

Gervini, Vitor. **Modelagem e controle de um servoposicionador pneumático via redes neurais**. Tese de Doutorado. Programa de pós-graduação em Engenharia Mecânica UFRGS, 2014.

Golnaraghi, Fraid; Kuo, Benjamin. **Automatic Control Systems**. Editora Wiley. Hoboken, New York. 2009.

Guenther, Raul; Perondi, Eduardo André. **O controle em cascata de sistemas pneumáticos de posicionamento**. Sociedade brasileira de Controle e Automação. 15, 149–161. 1999.

Guenther, Raul; Perondi, Eduardo. **Controle de um Servoposicionador Pneumático por Modos Deslizantes**. Congresso Nacional em Engenharia Mecânica CONEM 2000 15, 149–161. 2000.

Guez, A.; Ahmad, Z. **Solution to the inverse kinematics problem in robotics by neural networks**. In, IEEE International Conference on Neural Networks, pp. 617–624 vol.2. 1988.

Hama, Marcelo Tomio. **Uma plataforma orientada a agentes para o desenvolvimento de software em veículos aéreos não-tripulados**. Dissertação de mestrado. Programa de Pós-Graduação em Computação UFRGS, 2012.

IFTToMM - **International Federation for the Promotion of Mechanism and Machine Science**, 2015. Acessado em Julho de 2016. <http://www.iftomm2015.tw/>.

Intel® **Embedded Microcontrollers**. 2016. Acessado em Julho de 2016. <http://www.intel.com/content/www/us/en/homepage.html>.

Iordanou, H.N.; Surgenor, B.W. **Experimental evaluation of the robustness of discrete sliding mode control versus linear quadratic control**. IEEE Trans. Control Syst. Technol. 5, 254–260. 1997.

ISO 8373:2012 - **Robots and robotic devices -- Vocabulary. 2012**. Acessado em Julho de 2016. www.iso.org.

JADE Board. **Java Agent DEvelopment Framework**. 2016. Acessado em Julho de 2016. <http://jade.tilab.com/>.

Kunz, Guilherme de Oliveira. **Desenvolvimento de uma arquitetura programável de controle em tempo real para um servoposicionador pneumático**. Dissertação de mestrado. Programa de Pós-Graduação em Engenharia Mecânica. UFRGS, 2006.

Larizza, P.; Murciano, G.; Pappagallo, L.; Triggiani, G. **A new generation of modular robots**. pp. 3367–3371. 2006.

Maffei, Antonio; Onori, Mauro. **Evolvable Production Systems: Environment for New Business Models**. Key Eng. Mater. 467-469, 1592–1597. 2011.

Microchip, **Microchip Technology Inc.** 2016. Acessado em Julho de 2016. www.microchip.com.

Murata, S.; Yoshida, E.; Kamimura, A.; Kurokawa, H.; Tomita, K.; Kokaji, S.. **M-TRAN: self-reconfigurable modular robotic system**. IEEE/ASME Transactions on Mechatronics. vol 7, pag. 431–441. 2002.

NetBeans, **NetBeans IDE**. 2016. Acessado em julho de 2016. <https://netbeans.org>.

Niku, Saeed. **Introduction to Robotics**. Editora John Wiley & Sons. 2010.

Ogata, Katsuhiko. **Modern Control Engineering**. Editora Prentice Hall. 2010.

Oliveira, V.M.; Lages, W.F. **Linear Predictive Control of a Brachiation Robot**. In Canadian Conference on Electrical and Computer Engineering. pp. 1518–1521. 2006.

Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.-H. **Engineering Design**. Editora Springer London. 2007.

Pandian, S.R.; Takemura, F.; Hayakawa, Y.; Kawamura, S. **Pressure observer-controller design for pneumatic cylinder actuators**. IEEE/ASME Transactions on Mechatronics. vol 7, pp 490–499. 2002.

Pazos, Fernando. **Automação de sistema - robótica**. Editora Axel Books. 2002.

Peixoto, João Alvarez. **Desenvolvimento de sistemas de automação da manufatura usando arquiteturas orientadas a serviço e sistemas multi-agentes**. Dissertação de mestrado. Programa de Pós-Graduação em Engenharia Elétrica. UFRGS. 2012.

Perondi, Eduardo André. **Controle não-linear em cascata de um servoposicionador pneumático com compensação do atrito**. Tese de doutorado. Universidade Federal de Santa Catarina. 2002.

Posadas, Juan Luís. **Arquitectura para el control de robots móviles mediante delegación de código y agentes**. Tesis de doutorado. Universidad Politécnica de Valencia. 2004.

Poslad, Stefan. **Specifying Protocols for Multi-Agent Systems Interaction**. ACM Transactions on Autonomous and Adaptive Systems. vol 2. 2007.

Raspberry, **Teach, Learn, and Make with Raspberry Pi**. 2016. Acessado em Julho de 2016. www.raspberrypi.org.

RIA, **Robotic Industries Association**. 2016. Acessado em Julho de 2016. www.robotics.org.

Ribeiro, L.; Barata, J.; Onori, M.; Hanisch, C.; Hoos, J.; Rosa, R. **Self-organization in automation - the IDEAS pre-demonstrator**. In IECON 2011 - 37th Annual Conference on IEEE Industrial Electronics Society. pp. 2752–2757. 2011.

Ribeiro, Luís; Rosa, Rogerio; Cavalcante, André; Barata, José. **IADE–IDEAS agent development environment: lessons learned and research directions**. 4th CIRP Conference On Assembly Technologies And Systems, Michigan, USA. pp. 91–94. 2012.

Rijo, Marcos Giovane de Quevedo. **Desenvolvimento da base e controle do grau de liberdade rotacional de um robô cilíndrico com acionamento pneumático**. Dissertação de mestrado. Programa de Pós-Graduação em Engenharia Mecânica. UFRGS. 2013.

Rocha, André Dionísio Bettencourt da Silva. **An agent based architecture for material handling systems**. Dissertação de mestrado. Faculdade de Ciências e Tecnologia, Portugal. 2013.

Romano, Vitor Ferreira. **Robótica industrial: aplicação na indústria de manufatura e de processos**. Editora Edgard Blucher. ISBN 978-85-212-0315-5. 2002.

Ruggiero, Marcia; Lopes, Vera Lucia da Rocha. **Cálculo numérico: aspectos teóricos e computacionais**. Editora Makron Books do Brasil. ISBN 978-85-346-0204-4. 1996.

Russell, Stuart; Norvig, Peter. **Inteligência artificial**. Editora CAMPUS - RJ. ISBN 978-85-352-1177-1. 2004.

Santini, Diego Caberlon. **Arquitetura aberta para controle de robôs manipuladores**. Dissertação de mestrado. Programa de Pós-Graduação em Engenharia Elétrica. UFRGS. 2009.

Santos, Vitor. **Robótica Industrial**. Editora Universidade de Aveiro. 2003.

Sarmanho, Carlos Arthur Carvalho. **Desenvolvimento de um robô pneumático de 5 graus de liberdade com controlador não linear com compensação de atrito**. Tese de doutorado. Universidade Federal do Rio Grande do Sul. 2014.

Siciliano, Bruno; Sciavicco, Lorenzo; Villani, Luigi; Oriolo, Giuseppe. **Robotics: Modelling, Planning and Control**. Editora Springer Science & Business Media. 2010.

Slotine, J.J.E. **Putting physics in control-the example of robotics**. IEEE Control Systems Magazine. Vol 8, pp 12–18. 1988.

Sobczyk, Roland Mario Sobrinho. **Controle em cascata e a estrutura variável com adaptação de parâmetros e compensação de atrito de um servoposicionador pneumático**. Tese de doutorado. Universidade Federal do Rio Grande do Sul. 2009.

Spong, Mark; Hutchinson, Seth; Vidyasagar, M. **Robot Modeling and Control**. Editora Wiley. 2005.

Suzuki, Ricardo Murad. **Controle baseado em linearização por realimentação dos estados aplicado a um servoposicionador pneumático**. Dissertação de mestrado. Programa de Pós-Graduação em Engenharia Mecânica. UFRGS. 2010.

Ulewicz, S.; Schutz, D.; Vogel-Heuser, B. **Design, implementation and evaluation of a hybrid approach for software agents in automation**. In IEEE 17th Conference on Emerging Technologies Factory Automation (ETFA), pp. 1–4. 2012.

Vilanova, Ramon; Visioli, Antonio. **PID Control in the Third Millennium: Lessons Learned and New Approaches**. Editora Springer Science & Business Media. 2012.

Wooldridge, Michael. **An Introduction to MultiAgent Systems**. Editora John Wiley & Sons. ISBN 978-0-470-51946-2. 2009.

World Robotics. Acessado em julho de 2016. <http://www.worldrobotics.org>.

ANEXO A – ALGORITMOS DE CONTROLE

Aqui são descritos alguns dos controladores que podem ser utilizados nos módulos mecatrônicos aqui propostos, e já foram aplicados a sistemas de acionamento pneumáticos com resultados satisfatórios. Assim, estes algoritmos de controle destinados inicialmente aos dispositivos servoposicionadores, podem ser aplicados em cada grau de liberdade do robô modular conformado, já que cada uma das juntas pode ser controlada de forma individual. Conforme será apresentado, estes controladores não consideram a parcela dinâmica relativa a movimentação dos demais elos da cadeia cinemática serial, podendo gerar erros de seguimento de trajetória do sistema global. Nesta seção, serão apresentados os seguintes controladores: PID, realimentação de estados (PVA) e finalmente o controle por modos deslizantes (SMC). Cabe mencionar que existem outros tipos de controladores que foram utilizados com êxito para este tipo de sistemas, tais como apresentados por Perondi, 2002; Sarmanho, 2014; Sobczyk, 2009; Suzuki, 2010.

A.1 Controle PID

O controlador PID combina as ações de controle proporcional, integral e derivativa para gerar o sinal de controle. Este controlador alia os benefícios das três ações de controle que o compõem, fazendo com que seja possível eliminar em muitos sistemas erros de regime permanente, antecipar o comportamento do processo e acelerar a reação do sistema à presença de erro. Até os dias de hoje, este controlador é um dos mais comumente aplicado a sistemas industriais [Ogata, 2010].

Quando este controlador é aplicado a um servoposicionador, o erro de posição é integrado e derivado, sendo cada parcela multiplicada por um ganho respectivo para, através da soma destas ações, compor o sinal de controle $u(t)$, tal como apresentado na equação A.1.

$$u = K_p \left(e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{d(e(t))}{dt} \right). \quad \text{A.1}$$

Para a correta aplicação dos controladores PID, é necessário geralmente conhecer o comportamento do processo, para executar o ajuste adequado dos ganhos do controlador. Contudo, fatores como incertezas paramétricas e não-linearidades presentes em sistemas reais

dificultam a correspondência entre os modelos matemáticos obtidos através de equacionamentos de fenômenos físicos.

Controladores com ações do tipo Proporcional, Integral e Derivativo (PID) são largamente utilizados no cenário industrial devido ao fato de que são facilmente programáveis, versáteis e com capacidade de alterar o comportamento transitório e de regime permanente dos processos que se desejam controlar [Vilanova e Visioli, 2012]. São, porém, muito dependentes da amplitude dos ganhos associados às ações de controle e principalmente as condições iniciais do sistema. Entre os autores que já utilizaram este tipo de controlador em servoposicionadores pneumáticos, se encontram em Cukla, 2012; Perondi, 2002; Sarmanho, 2014; Sobczyk, 2009.

Através das considerações apresentadas pelos autores, conclui-se que, apesar deste controlador apresentarem características vantajosas, como uma elevada simplicidade operacional e facilidade de programação, os controladores PID expõem características que limitam sua aplicação ao controle de dispositivos acionados pneumaticamente. Dentre elas, ainda se destacam a alta sensibilidade às incertezas paramétricas, a baixa capacidade de compensação das não linearidades intrínsecas da planta e a limitação da localização dos polos em malha fechada, inviabilizando, ou ao menos, limitando sua aplicação ao controle de manipuladores robóticos acionados pneumaticamente, onde as exigências de precisão e rapidez de acionamento são elevadas.

A.2 Controle por realimentação de Estados

Na estratégia de controle por realimentação de estados, os valores dos estados do sistema que podem ser diretamente medidos ou matematicamente estimados são comparados com referências de forma a gerar um sinal de controle. Porém, para sua correta implementação é necessário que o sistema a ser controlado seja devidamente representado na forma de espaço de estados Golnaraghi e Kuo, 2009; Ogata, 2010. O uso deste método de controle é condicionado ao fato do sistema precisar ser controlável, ou seja, deve existir uma entrada de controle que a ele aplicada é capaz levar o conjunto de estados iniciais a um conjunto de estados em um intervalo de tempo finito.

Um dos motivos para aplicação deste controlador em manipuladores robóticos está relacionado à sua simplicidade de implementação computacional, sendo esta uma técnica de controle utilizada com sucesso no contexto de controle de sistemas pneumáticos. Alguns antecedentes nesta área é apresentado por Guenther e Perondi, 1999; Rijo, 2013 e Suzuki, 2010.

Quando a tarefa do controle de posição de um servossistema exige que não ocorra sobrepasso na resposta, o ganho proporcional que pode ser utilizado é ainda menor. De acordo com Guenther e Perondi, 1999, ao observar o tempo de assentamento de um servoposicionador, apresenta uma resposta lenta em malha fechada. Assim mesmo, os autores definem que essa dificuldade pode ser superada utilizando um modelo de 3ª ordem controlável através de uma realimentação de estados. Assim, ao menos teoricamente, os polos do sistema em malha fechada podem ser alocados em qualquer posição no plano complexo, o que possibilita obter a resposta especificada através de algum critério de projeto.

Estes controladores são referenciados como controladores PVA pois utilizam medidas de posição, velocidade e aceleração. A Figura A.1 mostra um diagrama de blocos para este controlador onde os polos do sistema em malha fechada são alocados pela escolha dos ganhos K_p (proporcional), K_v (de velocidade) e K_a (de aceleração) do controlador.

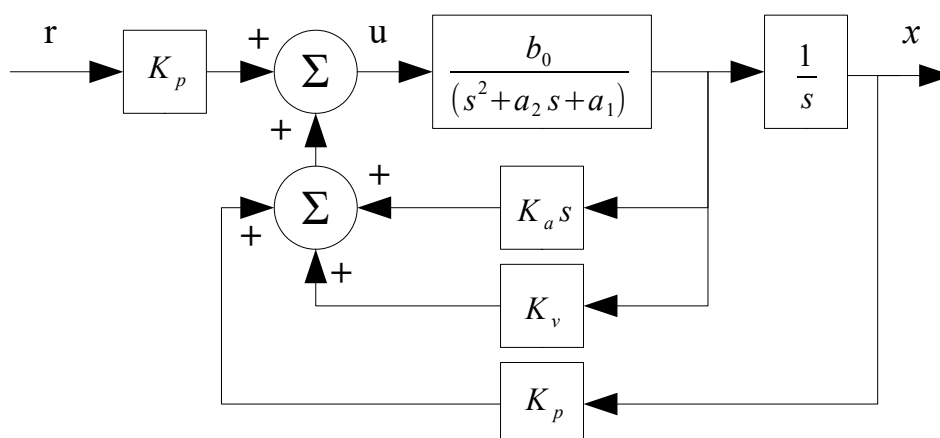


Figura A.1 - Modelo linear de terceira ordem com o controlador PVA

A aplicação de um controlador por realimentação de estados permite a utilização do método de alocação de polos para definição dos ganhos do controlador. Nesta técnica, os polos em malha fechada desejados podem ser determinados através de especificações de operação do sistema controlado, como características de regime transitório da resposta e/ou na resposta em frequência do sistema [Ogata, 2010]. O posicionamento dos polos, através deste método, requer além do fato de o sistema ser controlável, a necessidade do conhecimento de todos os estados (determinados por medição direta ou observação), sendo ainda necessário que os atuadores sejam capazes de impor uma dinâmica desejada, operando dentro de sua faixa linear de operação.

Cabe destacar que uma dificuldade prática para a implementação experimental do controlador PVA consiste na necessidade de dispor explicitamente do sinal referente a aceleração do sistema controlado, o que normalmente está associado à derivação numérica de sinais de velocidade e esquemas de filtragem, que por sua vez introduzem deslocamento de fase aos sinais.

A.3 Controladores por modos deslizantes

Os controladores com estrutura variável são conhecidos pela sua propriedade de alta robustez. Alguns autores, como Cukla, 2012; Iordanou e Surgenor, 1997; Pandian et al., 2002, tem proposto e estudado a aplicação de algoritmos de controle com estrutura variável para servoposicionadores pneumáticos.

A técnica de controle com estrutura variável mais utilizada é a baseada em modos deslizantes (*Slide Mode Control*). Basicamente, esta técnica consiste no chaveamento entre diferentes leis de controle com o propósito de manter o sistema na chamada superfície de deslizamento, que é especificada em projeto de acordo com a dinâmica desejada para o sistema em malha fechada. Em situação ideal, uma vez atingida esta superfície, o sistema mantém-se no regime de deslizamento ou em modo deslizante, permanecendo em equilíbrio dinâmico ao longo da trajetória dos estados definidos pela superfície. Para manter a trajetória de estados no regime de deslizamento, é necessária a aplicação de repetidas trocas de leis de controle para que a resposta se mantenha limitada pelo valor do erro especificado em projeto. Em sistemas reais este chaveamento ocorre em um tempo finito, o que pode dar origem a oscilações (fenômeno conhecido por *chattering*) que podem provocar desgastes nas peças móveis do atuador e introduzir ruído no sistema [Pandian et al., 2002; Guenther e Perondi, 2000].

A Figura A.2 exemplifica o *chattering* gerado pelas tentativas de manter o sistema em modo deslizante, onde y e \dot{y} são duas variáveis de estados (posição e velocidade), $S(x)$ é a trajetória dos estados e $S(x)=0$ denota a curva da trajetória de estados desejada (ou superfície de deslizamento).

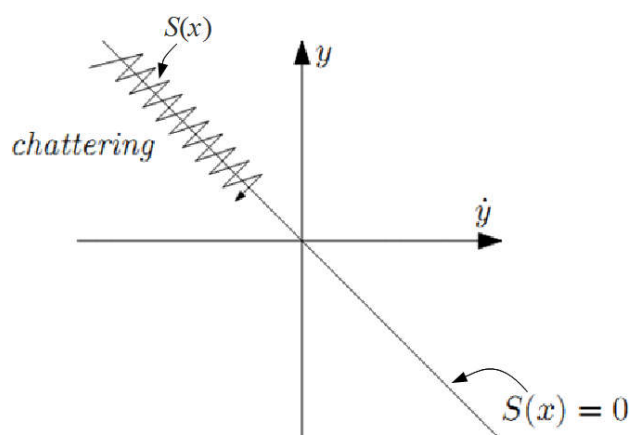


Figura A.2 – Fenômeno de *chattering* no controle de *slide mode*

O chattering, inerente a esta técnica de controle, somado ao ruído proveniente da derivação do sinal de velocidade para a obtenção da aceleração, consiste em uma das maiores dificuldades para a aplicação desta técnica para o controle de servoposicionadores pneumáticos [Guenther e Perondi, 2000].

ANEXO B – MÉTODOS ITERATIVOS DE NEWTON-RAPHSON

De acordo com Chapra e Canale, 2008, um método iterativo consiste em uma sequência de instruções passo a passo que normalmente são executadas ciclicamente, onde a execução de um ciclo recebe o nome de iteração.

Os métodos iterativos são métodos numéricos utilizados para a resolução de funções reais, cujo objetivo é encontrar soluções de equações não lineares do tipo $f(x) = 0$, tais como a resolução das equações da cinemática inversa de um robô.

O Método de Newton-Raphson é um caso particular do método de iteração linear. Um método de iteração linear consiste em estimar a raiz de uma função $f(x)$ usando o processo iterativo:

$$x_n = \varphi(x_n). \quad (\text{B.1})$$

Pode-se escrever uma forma geral para a função de iteração:

$$\varphi(x) = x + A(x)f(x), \quad (\text{B.2})$$

pois, para x igual à raiz de $f(x) = 0$, ou seja $x = \varphi(x)$ para qualquer $A(x) \neq 0$.

Para haver a convergência no método da iteração linear é preciso que $|\varphi'(x)| < 1$ em um intervalo $[a, b]$ que contem a raiz de $f(x)$ e $\alpha \in [a, b]$. Com isso, tem-se $\varphi'(x) < 1$ desde que não se afaste muito do valor de α durante o processo de resolução do problema.

Derivando $\varphi(x)$ dada pela Equação D.2 em relação a x , tem-se:

$$\varphi'(x) = 1 + A'(x)f(x) + A(x).f'(x). \quad (\text{B.3})$$

Calculando em $x = \alpha$, segue que:

$$\varphi'(\alpha) = 1 + A(\alpha)f'(\alpha). \quad (\text{B.4})$$

Forçando a condição $\varphi'(x) = 0$, tem-se:

$$A(\alpha).f'(\alpha) = 1, \quad (\text{B.5})$$

ou seja,

$$A(\alpha) = \frac{1}{f'(\alpha)}. \quad (\text{B.6})$$

Escolhendo:

$$A(x) = \frac{1}{f'(x)}, \quad (\text{B.7})$$

segue da Equação B.2 que:

$$\varphi(x) = x - \frac{f(x)}{f'(x)}. \quad (\text{B.8})$$

O método de Newton-Raphson consiste em usar o processo iterativo $x_{n+1} = \varphi(x_n)$ e como função de iteração a Equação B.8.

B.1 Convergência do Método de Newton-Raphson

Apesar de obter-se a forma da função $\varphi(x)$ procurando garantir a convergência do processo iterativo, esta não está sempre garantida para este método. A convergência no método de Newton-Raphson está sempre garantida para um certo intervalo $[a, b]$ que contém a raiz de $f(x)$, desde que $f(x)$ e $f'(x)$ sejam contínuas nesse intervalo e que $f'(\alpha) \neq 0$, onde α é a raiz de $f(x)$ ($f(\alpha) = 0$). Portanto, se for utilizada uma estimativa inicial x_0 tal que $x_0 \in [a, b]$, a convergência estará garantida. Em outras palavras, para o método de Newton-Raphson convergir, é preciso que nossa estimativa inicial esteja próxima da raiz de $f(x)$. A proximidade exigida para a convergência vai depender de caso a caso e nem sempre é simples de determinar.

B.2 Interpretação geométrica

Dado x_n , o ponto x_{n+1} será obtido pela intercessão da reta tangente a $f(x)$ em x_n , com o eixo x (a abscissa). Pode-se ilustrar isso matematicamente. A reta tangente a $f(x)$ em x_n , é dada por:

$$f'(x) = \frac{f(x_n) - 0}{x_n - x_{n-1}}. \quad (\text{B.9})$$

A partir dessa expressão, é obtida a fórmula de Newton-Raphson, ou seja:

$$x_{n-1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (\text{B.10})$$

Portanto, a cada iteração do nosso processo, aproxima-se cada vez mais da raiz de $f(x)$ através da tangente (ou seja, da derivada) da função $f(x)$.

A Figura B.1 a seguir ilustra essa interpretação geométrica do Método de Newton-Raphson.

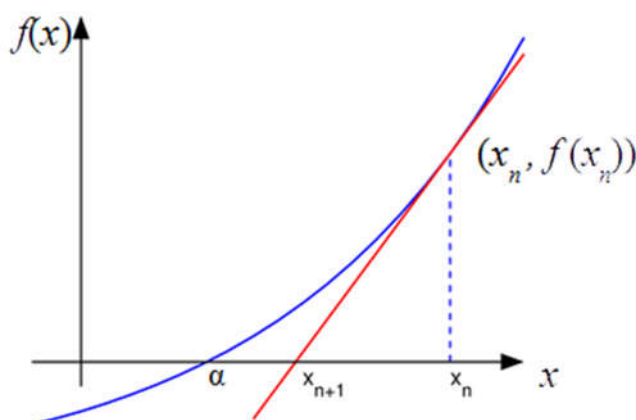


Figura B.1 – Interpretação geométrica do método de Newton-Raphson [Chapra e Canale, 2008]